Migration

SAP Sybase IQ 16.0 SP04

UNIX/Linux

DOCUMENT ID: DC01778-01-1604-01
LAST REVISED: May 2014

Contents

Support Package Installation Workflow 1

Upgrading to 16.0 SP04 3

Preparing for Database 15.x Upgrades 4

Preparing an SAP Sybase IQ Server 4

Installing SAP Sybase IQ 16.0 SP04 5

Upgrading SAP Sybase IQ Databases to 16.0 SP04 6

Regrant EXECUTE Privilege After Upgrading a Database 7

Logical Servers 8

Upgrading to 16.0 SP04 and SCC 3.3 9

Preparing to Install SAP Sybase IQ 16.0 SP04 with SAP Control Center 10

Preparing an SAP Sybase IQ Server 10

Preparing an SAP Control Center Server 11

Preparing for Database 15.x Upgrades 12

Installing 16.0 SP04 on a Shared IQ and SCC Host 12

Installing SAP Sybase IQ 16.0 SP04 on a Standalone SCC Server 14

Upgrading an SAP Sybase IQ Database Using Interactive SQL 14

Upgrading a Database Using the SCC Upgrade Database Wizard 15

Regrant EXECUTE Privilege After Upgrading a Database 15

Considerations When Upgrading from Pre-16.0 Releases 17

Space Considerations 17

System Procedures Changes After Upgrading from 15.x 17

Pre-16.0 Privileged System Procedures 18

Role-Based Security After Upgrading from 15.x 20

What Happened to Authorities, Permissions, and Groups? 20
Authorities Become Compatibility Roles 21
Permissions Become Privileges 23
Groups Become Roles 23
Change to Concept of a Super-User (DBA Authority) 24
Changes to the GRANT Statement Syntax 25
Changes to the REVOKE Statement Syntax 27
Changes to REMOTE DBA 29
Changes in Inheritance Behavior 29
Changes in Administering the Database Publisher 30
Compatibility Roles 30
Backward Compatibility in SAP Sybase IQ 16.0 57
Stored Procedure to Map Authorities to System Roles 57
Connecting to SAP Sybase IQ 15.x Databases with SAP Sybase IQ 16.0 57
SAP Sybase IQ 12.7 Database Migration 57
Preparing to Migrate 57
Migration Utilities 58
Migration Issues 64
Unloading Legacy Schemas 71
Migrating Legacy Databases 72
Postmigration Tasks 85
Hardware Changes 90
Moving 32-Bit Databases to 64-bit Platforms 90
Converting to a New Hardware Platform 91
Index 93

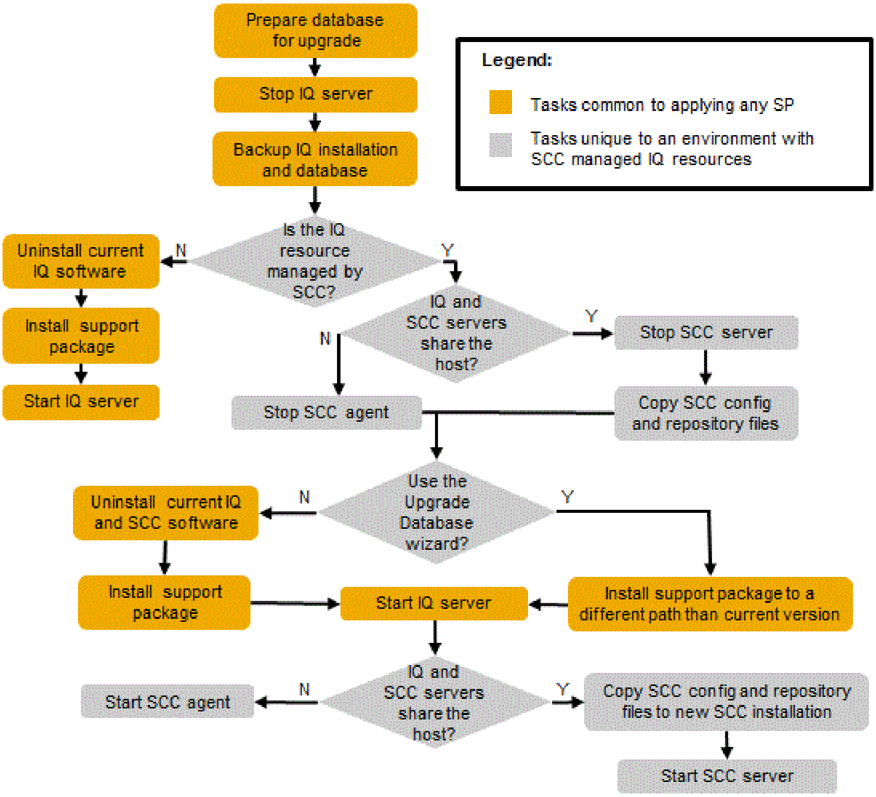
Support Package Installation Workflow

The workflow to apply an SAP® Sybase® IQ support package (SP) depends on whether you are using SAP® Control Center to manage the SAP Sybase IQ resources.

If you are not using SCC, simply stop the IQ server, install the support package, and restart the server.

However, if you have SAP Sybase IQ-managed resources, there are additional considerations, for example whether SAP Sybase IQ and SAP Control Center share the same host, or if you are migrating from 15.3 or 15.4, whether you plan to use the SCC Upgrade Database Wizard. Such considerations require additional tasks.

The workflow highlights the common tasks and the different decision points when IQ managed resources are involved.


Prepare database for upgrade
Stop IQ server
Backup IQ installation and database
Legend:
Tasks common to applying any SP
Tasks unique to an environment with SCC managed IQ resources
Is the IQ resource managed by SCC?
N
Y
Uninstall current IQ software
Install support package
Start IQ server
IQ and SCC servers share the host?
N
Y
Stop SCC server
Copy SCC config and repository files
Stop SCC agent
Use the Upgrade Database wizard?
N
Y
Uninstall current IQ and SCC software
Install support package
Start IQ server
Install support package to a different path than current version
IQ and SCC servers share the host?
N
Y
Start SCC agent
Copy SCC config and repository files to new SCC installation
Start SCC server

If you are not using SAP Control Center to manage your SAP Sybase IQ resources, refer to the tasks under *Upgrading to 16.0 SP04*.

If you are using SAP Control Center to manage your SAP Sybase IQ resources, refer to the tasks under *Upgrading to 16.0 SP04 and SCC*.

See also

- *Upgrading to 16.0 SP04* on page 3
- *Upgrading to 16.0 SP04 and SCC 3.3* on page 9

Upgrading to 16.0 SP04

SAP® Sybase® IQ 16.0 SP04 includes updates to features that are currently installed on your system. See the SAP Sybase IQ 16.0 SP04 cover letter, included as part of this release, for a list of updates.

SAP Sybase IQ 16.0 SP04 includes SAP Control Center 3.3.

You cannot manage a 16.0 SP04 resource with previous versions of SAP Control Center. However, you can use SCC 3.3 to manage all previous versions of SAP Sybase IQ from 15.3.

To administer an SAP Sybase IQ 16.0 SP04 database with SCC, you must install an SCC 3.3 server and an SCC 3.3 Agent for SAP Sybase IQ.

SCC installation options depend on whether SCC 3.2 is currently installed or you whether you are installing SCC for the first time. For general SCC installation instructions and deployment strategies, see the section *SAP Control Center* in the *Installation and Configuration Guide*. For release-specific instructions and migration tips, see the *SAP Control Center 3.3* section in this document.

1. *Preparing for Database 15.x Upgrades*

   SAP Sybase IQ 16.0 SP04 includes changes to the system tables and database schema. If you are migrating a SAP Sybase IQ 15.x database, objects no longer supported by SAP Sybase IQ 16.0 must be removed or modified.
2. *Preparing an SAP Sybase IQ Server*

   SAP Sybase IQ 16.0 SP04 is a complete installation package. The initial release of SAP Sybase IQ 16.0 is not a prerequisite for this support package.
3. *Installing SAP Sybase IQ 16.0 SP04*

   Some steps may differ for simplex and multiplex servers when installing SAP Sybase IQ 16.0 SP04. See the cover letter for release-specific instructions.
4. *Upgrading SAP Sybase IQ Databases to 16.0 SP04*

   Upgrade your database to SAP Sybase IQ 16.0 SP04 before you perform database read-write operations.
5. *Regrant EXECUTE Privilege After Upgrading a Database*

   When you upgrade a database, there is a set of privileged system procedures for which you must regrant the EXECUTE privilege if the upgraded database continues to use the SYSTEM PROCEDURE DEFINER security model.
6. *Logical Servers*

   An SAP Sybase IQ 16.0 multiplex database upgrade changes the way users access multiplex servers. Starting with SAP Sybase IQ 15.4, logical servers provide the only means to access the multiplex server nodes.

See also

- *Upgrading to 16.0 SP04 and SCC 3.3* on page 9
- *Support Package Installation Workflow* on page 1

Preparing for Database 15.x Upgrades

SAP Sybase IQ 16.0 SP04 includes changes to the system tables and database schema. If you are migrating a SAP Sybase IQ 15.x database, objects no longer supported by SAP Sybase IQ 16.0 must be removed or modified.

Note: All steps are specific to customers who are upgrading from SAP Sybase IQ 15.x. Customers upgrading from SAP Sybase IQ 16.0 or later can skip these steps.

1. Disconnect all users from the 15.x server.
2. Drop all `JOIN` and all `LD` indexes from the 15.x database.
3. Update `DATE` columns that contain a time portion.

   There is a known issue that affects any partition defined on a `DATE` column that contains a time portion:
   - Use **ALTER TABLE MERGE** to combine the partition with the next partition.
   - Use **ALTER TABLE SPLIT** to divide the resulting partition into a definition with the same criteria, but no time portion.
4. Drop and re-create all SAP Sybase IQ 15.2 `TEXT` indexes that have not already been dropped and re-created as part of a version 15.2 ESD upgrade.

   `TEXT` indexes created in SAP Sybase IQ 15.2 are incompatible with later versions of SAP Sybase IQ.
5. Shut down the SAP Sybase IQ 15.x server or multiplex.

   If the server stops responding during shutdown, do not proceed to the next step. Restart the database and shut it down again. Proceed to the next task only on a clean shutdown.

Preparing an SAP Sybase IQ Server

SAP Sybase IQ 16.0 SP04 is a complete installation package. The initial release of SAP Sybase IQ 16.0 is not a prerequisite for this support package.

1. Download the SAP Sybase IQ 16.0 SP04 archive.

If you purchased the product directly from Sybase	Download the support package from the Software Downloads for EBFs and Maintenance site at: *http://downloads.sybase.com/swd/base.do?client=support* For details about SAP Sybase software downloads, see Software Downloads Frequently Asked Questions at: *http://downloads.sybase.com/swd/jsp/faq.jsp*.
If you ordered your Sybase product under an SAP contract	Download the support package from the SAP Software Download Center at: *http://service.sap.com/swdc* For details, see the online help on the SAP Software Download Center Web site.

2. Back up your current installation, and save copies of any changes you made to default login and post-login scripts. Before you proceed, make sure the backups are readable.
3. Check with your operating system vendor for information about the latest operating system patches. Use the recommended operating system patch. Do not use a patch that is earlier than the version suggested for your operating system.
4. In Interactive SQL, run **sp_iqcheckoptions** on each database, and capture the output.

   sp_iqcheckoptions generates a list of current database values and options. Use these values to restore your database settings after you upgrade.
5. Validate your license against each database to ensure that your license maintenance support is in (or near) compliance.

   A server without a maintenance contract for more than a year will not run. Validating your licenses ensures that your database functions correctly after you install the SP. See the *SySAM documentation* for validation procedures.

Installing SAP Sybase IQ 16.0 SP04

Some steps may differ for simplex and multiplex servers when installing SAP Sybase IQ 16.0 SP04. See the cover letter for release-specific instructions.

The SAP Sybase IQ 16.0 SP04 installer includes SAP Control Center 3.3. SAP Sybase IQ 16.0 SP04 is not compatible with previous versions of SCC. To administer an SAP Sybase IQ 16.0 SP04 database with SCC, you must install an SCC 3.3 server and an SCC 3.3 Agent.

1. Use the appropriate utility to extract the SAP Sybase IQ 16.0 SP04 archive to a directory on the file system.
2. For:

Server	Action
Simplex	Shut down the server.

Server	Action
Multiplex	Shut down all multiplex nodes.

For server shutdown instructions, see *Administration: Database > Run SAP Sybase IQ Database Servers > Database Server Shutdown > Stopping Servers*.

3. Install SAP Sybase IQ 16.0 SP04.

   On multiplex servers, update the software on each node separately.
4. In a terminal, change to `$IQDIR16/bin64`.
5. Check the server version string:

   ```
   start_iq -v2
   ```

   Verify that the results match the version string in the cover letter.
6. Restart the server or multiplex.

Upgrading SAP Sybase IQ Databases to 16.0 SP04

Upgrade your database to SAP Sybase IQ 16.0 SP04 before you perform database read-write operations.

1. For:

Server	Action
Simplex	Start the SAP Sybase IQ 16.0 SP04 server. Use the **-gm 1** and **-iqro 1** startup flags.
Multiplex	Start the coordinator on the SAP Sybase IQ 16.0 SP04 multiplex. Use the **-iqmpx_sn 1**, **-gm 1**, and **-iqro 1** startup flags.

   The **-gm** switch controls the number of concurrent connections.
2. Start Interactive SQL and connect to the database.
3. Upgrade the database.

   The **ALTER DATABASE UPGRADE** instantiates all SAP Sybase IQ 16.0 SP04 binaries and schema changes. To roll back to a previous version, you must restore from backup.
4. Run **sp_iqcheckdb ('allocation database')** to verify that there are no errors.
5. For:

Server	Action
Simplex	Shut down and restart the server normally (without the **-gm 1** and **-iqro 1** startup flags).

Server	Action
Multiplex	Shut down and restart the coordinator normally (without the **-iqmpx_sn 1**, **-gm 1**, and **-iqro 1** startup flags). Synchronize and restart all multiplex secondary servers.

6. Back up the database.

Regrant EXECUTE Privilege After Upgrading a Database

When you upgrade a database, there is a set of privileged system procedures for which you must regrant the EXECUTE privilege if the upgraded database continues to use the SYSTEM PROCEDURE DEFINER security model.

The system procedures for which you must regrant the privilege are:

- sa_audit_string
- sa_checkpoint_execute
- sa_disable_auditing_type
- sa_disk_free_space
- sa_enable_auditing_type
- sa_external_library_unload
- sa_flush_cache
- sa_list_external_library
- sa_server_option
- sa_procedure_profile
- sa_procedure_profile_summary
- sa_table_page_usage
- sa_validate

To determine the security model of your upgraded database, execute:

```
select IF ((HEXTOINT(substring(db_property('Capabilities'),
1,length(db_property('Capabilities'))-20)) & 8) = 8)
THEN 1
ELSE 0
END IF
```

A value of 1 indicates the database is using the SYSTEM PROCEDURE INVOKER model. A value of 0 indicates that the database is using the SYSTEM PROCEDURE DEFINER.

The MANAGE ANY OBJECT PRIVILEGE system privilege is required to grant the EXECUTE object-level privilege.

To grant the privilege, execute:

```
GRANT EXECUTE
   ON [ owner.] {procedure-name | user-defined-function-name }
   TO userID [,...]
```

See also

- *Pre-16.0 Privileged System Procedures* on page 18

Logical Servers

An SAP Sybase IQ 16.0 multiplex database upgrade changes the way users access multiplex servers. Starting with SAP Sybase IQ 15.4, logical servers provide the only means to access the multiplex server nodes.

Upgrading a multiplex database creates an appropriate logical server for each server-specific login policy. Login policies are updated to use a logical server configuration that provides access to the same set of multiplex servers as prior to upgrade.

If a login policy does not allow access to any node (such as when the base setting of `LOCKED` is `ON` and there are no multiplex server-level overrides), the login policy is set to a system-defined logical server, `NONE`, instead of creating a new logical server. `NONE` indicates that the login policy does not allow access to any multiplex server.

If a login policy has no explicit setting for the `LOCKED` option, either at the base level or via a multiplex server-level override, no logical server is created for that policy. Such a login policy inherits the logical server assignment of the root login policy.

- Membership configuration of a logical server provides access to the same multiplex nodes as the corresponding 15.x login policy. If the login policy allowed access to the current coordinator server, a logical membership of the coordinator is also added to the logical server.
- Logical server names are derived from login policy names. If the login policy name is fewer than 126 characters, the logical server follows this naming convention: **LS_<login policy name>**. For example, for a login policy named **mpx_grp1**, a logical server **LS_mpx_grp1** is created and assigned to that login policy.

  If the login policy name exceeds 125 characters, a logical server is created with the same name as of the login policy, that is, without adding an **LS_** prefix.
- During the upgrade, some login policy option settings or multiplex server-level overrides are reset or removed. In the root login policy, `LOCKED` and `MAX_CONNECTIONS` overrides are reset to default values ('`OFF`' and 10 respectively).

  Settings for `LOCKED` and `MAX_CONNECTIONS` are removed from user-defined login policies. Multiplex server-level overrides are removed from all login policies.
- To retain pre-upgrade behavior for existing applications, the login policy option `LOGIN_REDIRECTION` is added to the root logical server policy with its value set to off.

Note: See *Administration: Multiplex > Manage Resources Through Logical Servers*.

Upgrading to 16.0 SP04 and SCC 3.3

If you currently use SAP Control Center to manage SAP Sybase IQ resources, and are installing SAP Sybase IQ 16.0 SP04, you must upgrade the SCC server before you upgrade the managed resources. See the SAP Sybase IQ 16.0 SP04 cover letter, included as par of this release, for a list of updates.

SAP Sybase IQ 16.0 SP04 includes SAP Control Center 3.3. SAP Control Center 3.3 is required to administer an SAP Sybase IQ 16.0 SP04 database. SCC 3.3 is also compatible with all previous versions of SAP Sybase IQ from 15.3.

For general SCC installation instructions and deployment strategies for a new SAP Control Center installation, see the SAP Control Center section in the *Installation and Configuration Guide*.

1. *Preparing to Install SAP Sybase IQ 16.0 SP04 with SAP Control Center*

   There are several tasks that you must perform before installing SAP Sybase IQ 16.0 SP04 in an SAP Sybase IQ environment managed by SAP Control Center.
2. *Installing 16.0 SP04 on a Shared IQ and SCC Host*

   Some steps may differ for simplex and multiplex servers when installing SAP Sybase IQ 16.0 SP04. See the cover letter for release-specific instructions.
3. *Installing SAP Sybase IQ 16.0 SP04 on a Standalone SCC Server*

   Migrate a standalone SAP Control Center server to 3.3.
4. *Upgrading an SAP Sybase IQ Database Using Interactive SQL*

   Before you perform database read-write operations, upgrade your 15.x or 16.x database to SAP Sybase IQ 16.0 SP04.
5. *Upgrading a Database Using the SCC Upgrade Database Wizard*

   The SAP Control Center Upgrade Database wizard lets you upgrade an SAP Sybase IQ 15.3 or 15.4 simplex or multiplex database to SAP Sybase IQ 16.0 SP04.
6. *Regrant EXECUTE Privilege After Upgrading a Database*

   When you upgrade a database, there is a set of privileged system procedures for which you must regrant the EXECUTE privilege if the upgraded database continues to use the SYSTEM PROCEDURE DEFINER security model.

See also

- *Upgrading to 16.0 SP04* on page 3
- *Support Package Installation Workflow* on page 1

Preparing to Install SAP Sybase IQ 16.0 SP04 with SAP Control Center

There are several tasks that you must perform before installing SAP Sybase IQ 16.0 SP04 in an SAP Sybase IQ environment managed by SAP Control Center.

1. *Preparing an SAP Sybase IQ Server*

   SAP Sybase IQ 16.0 SP04 is a complete installation package. The initial release of SAP Sybase IQ 16.0 is not a prerequisite for this support package.
2. *Preparing an SAP Control Center Server*

   There are several task you must perform on the on the SAP Control Center server before you install SAP Sybase IQ 16.0 SP04.
3. *Preparing for Database 15.x Upgrades*

   SAP Sybase IQ 16.0 SP04 includes changes to the system tables and database schema. If you are migrating a SAP Sybase IQ 15.x database, objects no longer supported by SAP Sybase IQ 16.0 must be removed or modified.

See also

- *Installing 16.0 SP04 on a Shared IQ and SCC Host* on page 12

Preparing an SAP Sybase IQ Server

SAP Sybase IQ 16.0 SP04 is a complete installation package. The initial release of SAP Sybase IQ 16.0 is not a prerequisite for this support package.

1. Download the SAP Sybase IQ 16.0 SP04 archive.

If you purchased the product directly from Sybase	Download the support package from the Software Downloads for EBFs and Maintenance site at: *http://downloads.sybase.com/swd/base.do?client=support* For details about SAP Sybase software downloads, see Software Downloads Frequently Asked Questions at: *http://downloads.sybase.com/swd/jsp/faq.jsp*.
If you ordered your Sybase product under an SAP contract	Download the support package from the SAP Software Download Center at: *http://service.sap.com/swdc* For details, see the online help on the SAP Software Download Center Web site.

2. Back up your current installation, and save copies of any changes you made to default login and post-login scripts. Before you proceed, make sure the backups are readable.

3. Check with your operating system vendor for information about the latest operating system patches. Use the recommended operating system patch. Do not use a patch that is earlier than the version suggested for your operating system.
4. In Interactive SQL, run **sp_iqcheckoptions** on each database, and capture the output.

   sp_iqcheckoptions generates a list of current database values and options. Use these values to restore your database settings after you upgrade.
5. Validate your license against each database to ensure that your license maintenance support is in (or near) compliance.

   A server without a maintenance contract for more than a year will not run. Validating your licenses ensures that your database functions correctly after you install the SP. See the *SySAM documentation* for validation procedures.

Preparing an SAP Control Center Server

There are several task you must perform on the on the SAP Control Center server before you install SAP Sybase IQ 16.0 SP04.

Note: If SAP Control Center and SAP Sybase IQ reside on the same host, skip step 1.

1. Back up your current installation. Before you proceed, make sure the backup is readable.
2. Make a copy of the `csi_config.xml` file in the `$SYBASE/SCC-3_2/conf` directory.

   Note: This file contains the SCC login IDs and passwords and is overwritten during migration.
3. Make a copy of these repository files:
   - `$SYBASE/SCC-3_2/services/Repository/scc_repository.db`
   - `$SYBASE/SCC-3_2/services/Repository/scc_repository.log`
   - `$SYBASE/SCC-3_2/services/Repository/service-config.xml`
4. Make a copy of these files:
   - `$SYBASE/SCC-3_2/services/SccSADataserver/service-config.xml`
   - `$SYBASE/SCC-3_2/services/EmbeddedWebContainer/service-config.xml`

   Warning! You are copying multiple instances of the `service-config.xml` file that are stored in different locations. Though the file name is common, the content of each instance is unique. When making each copy, note its original location, as you will need this information to complete the migration process. Store each copy in a different location to avoid accidental overwrites.

Preparing for Database 15.x Upgrades

SAP Sybase IQ 16.0 SP04 includes changes to the system tables and database schema. If you are migrating a SAP Sybase IQ 15.x database, objects no longer supported by SAP Sybase IQ 16.0 must be removed or modified.

Note: All steps are specific to customers who are upgrading from SAP Sybase IQ 15.x. Customers upgrading from SAP Sybase IQ 16.0 or later can skip these steps.

1. Disconnect all users from the 15.x server.
2. Drop all `JOIN` and all `LD` indexes from the 15.x database.
3. Update `DATE` columns that contain a time portion.

   There is a known issue that affects any partition defined on a `DATE` column that contains a time portion:
   - Use **ALTER TABLE MERGE** to combine the partition with the next partition.
   - Use **ALTER TABLE SPLIT** to divide the resulting partition into a definition with the same criteria, but no time portion.
4. Drop and re-create all SAP Sybase IQ 15.2 `TEXT` indexes that have not already been dropped and re-created as part of a version 15.2 ESD upgrade.

   `TEXT` indexes created in SAP Sybase IQ 15.2 are incompatible with later versions of SAP Sybase IQ.
5. Shut down the SAP Sybase IQ 15.x server or multiplex.

   If the server stops responding during shutdown, do not proceed to the next step. Restart the database and shut it down again. Proceed to the next task only on a clean shutdown.

Installing 16.0 SP04 on a Shared IQ and SCC Host

Some steps may differ for simplex and multiplex servers when installing SAP Sybase IQ 16.0 SP04. See the cover letter for release-specific instructions.

The SAP Sybase IQ 16.0 SP04 installer includes SAP Control Center 3.3. SAP Sybase IQ 16.0 SP04 is not compatible with previous versions of SCC. To administer an SAP Sybase IQ 16.0 SP04 database with SCC, you must install an SCC 3.3 server and an SCC 3.3 Agent.

The SAP Control Center Upgrade Database wizard lets you upgrade an SAP Sybase IQ 15.3 or 15.4 simplex or multiplex database to SAP Sybase IQ 16.0 SP04. The Upgrade Database wizard does not currently support the upgrade of an earlier 16.0 version to 16.0 SP04.

To use the Upgrade Database wizard, you must have both the 15.x and 16.0 SP04 versions of the SAP Sybase IQ server software installed on the same machine. Once the database upgrade is complete, uninstall the earlier version of the SAP Sybase IQ server software.

1. Use the appropriate utility to extract the SAP Sybase IQ 16.0 SP04 archive to a directory on the file system.
2. For:

Server	Action
Simplex	Shut down the server.
Multiplex	Shut down all multiplex nodes.

   For server shutdown instructions, see *Administration: Database > Run Database Servers > Ways to Start and Stop Databases*.

3. For:

Configuration	Action
SCC and SAP Sybase IQ servers reside on the same host.	Shut down the SCC 3.2 server.
SAP Sybase IQ is managed by a remote SCC server.	Shut down the SCC 3.2 agent.

4. If you do not plan to use the Upgrade Database Wizard to upgrade your 15.3 or 15.4 database, uninstall the current SAP Sybase IQ server.
5. Install SAP Sybase IQ 16.0 SP04.
   - If SAP Sybase IQ 15.x is still installed, specify a different path. Do not overwrite the existing version.
   - On multiplex servers, update the software on the each node separately.
6. In a terminal, change to `$IQDIR16/bin64`.
7. Check the server version string:

```
start_iq  -v2
```

   Verify that the results match the version string in the cover letter.

8. Restart the server or multiplex.
9. Copy the saved SCC `csi_config.xml` file into the `$SYBASE/SCC-3_3/conf` directory, overwriting the existing file.
10. Copy the saved SCC repository files into the `$SYBASE/SCC-3_3/services/Repository` directory, overwriting the existing files.
11. Copy the `service-config.xml` file saved from the original `SccSADataserver` location to the corresponding new `$SYBASE/SCC-3_3/services/SccSADataserver/` location, overwriting the existing copy.
12. Copy the `service-config.xml` file saved from the original `EmbeddedWebContainer` location to the corresponding new `$SYBASE/SCC-3_3/services/EmbeddedWebContainer/` location, overwriting the existing copy.

13. For:

Configuration	Action
SCC and SAP Sybase IQ servers reside on the same host.	Start the SCC 3.3 server.
SAP Sybase IQ is managed by a remote SCC server.	Start the SCC 3.3 agent.

See also

- *Preparing to Install SAP Sybase IQ 16.0 SP04 with SAP Control Center* on page 10

Installing SAP Sybase IQ 16.0 SP04 on a Standalone SCC Server

Migrate a standalone SAP Control Center server to 3.3.

1. Shut down and uninstall the SCC 3.2 server.
2. Install the SAP Control Center 3.3 server.

   Warning! Adjust the installation options to install the SCC server only.
3. Copy the backedup SCC `csi_config.xml` file into the `$SYBASE/SCC-3_3/conf` directory, overwriting the existing file.
4. Copy the saved SCC repository files into the `$SYBASE/SCC-3_3/services/Repository` directory, overwriting the existing files.
5. Copy the `service-config.xml` file saved from the original `SccSADataserver` location to the corresponding new `$SYBASE/SCC-3_3/services/SccSADataserver/` location, overwriting the existing copy.
6. Copy the `service-config.xml` file saved from the original `EmbeddedWebContainer` location to the corresponding new `$SYBASE/SCC-3_3/services/EmbeddedWebContainer/` location, overwriting the existing copy.
7. Start the SAP Control Center 3.3 server.

Upgrading an SAP Sybase IQ Database Using Interactive SQL

Before you perform database read-write operations, upgrade your 15.x or 16.x database to SAP Sybase IQ 16.0 SP04.

1. For:

Server	Action
Simplex	Start the SAP Sybase IQ 16.0 SP04 server. Use the **-gm 1** and **-iqro 1** startup flags.
Multiplex	Start the coordinator on the SAP Sybase IQ 16.0 SP04 multiple. Use the **-iqmpx_sn 1**, **-gm 1**, and **-iqro 1** startup flags.

The **-gm** switch controls the number of concurrent connections.

2. Start Interactive SQL and connect to the database.
3. Upgrade the database:

   ALTER DATABASE UPGRADE instantiates all SAP Sybase IQ 16.0 SP04 binaries and schema changes. To roll back to a previous version, you must restore from backup.
4. Run **sp_iqcheckdb ('allocation database')** to verify that there are no errors.
5. For:

Server	Action
Simplex	Shut down and restart the server normally (without the **-gm 1** and **-iqro 1** startup flags).
Multiplex	Shut down and restart the coordinator normally (without the **-iqmpx_sn 1**, **-gm 1**, and **-iqro 1** startup flags). Synchronize and re-start all multiplex secondary servers.

6. Back up the database.

Upgrading a Database Using the SCC Upgrade Database Wizard

The SAP Control Center Upgrade Database wizard lets you upgrade an SAP Sybase IQ 15.3 or 15.4 simplex or multiplex database to SAP Sybase IQ 16.0 SP04.

For details on using the Upgrade Database wizard, see *Upgrading a Simplex Database* and *Upgrading a Multiplex Database* in the *SAP Control Center for SAP Sybase IQ documentation*.

Regrant EXECUTE Privilege After Upgrading a Database

When you upgrade a database, there is a set of privileged system procedures for which you must regrant the EXECUTE privilege if the upgraded database continues to use the SYSTEM PROCEDURE DEFINER security model.

The system procedures for which you must regrant the privilege are:

- sa_audit_string
- sa_checkpoint_execute
- sa_disable_auditing_type
- sa_disk_free_space
- sa_enable_auditing_type
- sa_external_library_unload
- sa_flush_cache
- sa_list_external_library
- sa_server_option
- sa_procedure_profile
- sa_procedure_profile_summary
- sa_table_page_usage
- sa_validate

To determine the security model of your upgraded database, execute:

```
select IF ((HEXTOINT(substring(db_property('Capabilities'),
1,length(db_property('Capabilities'))-20)) & 8) = 8)
THEN 1
ELSE 0
END IF
```

A value of 1 indicates the database is using the SYSTEM PROCEDURE INVOKER model. A value of 0 indicates that the database is using the SYSTEM PROCEDURE DEFINER.

The MANAGE ANY OBJECT PRIVILEGE system privilege is required to grant the EXECUTE object-level privilege.

To grant the privilege, execute:

```
GRANT EXECUTE
  ON [ owner.] {procedure-name | user-defined-function-name }
  TO userID [,...]
```

See also

- *Pre-16.0 Privileged System Procedures* on page 18

Considerations When Upgrading from Pre-16.0 Releases

Upgrading from pre-16.0 releases has additional implications, including the role-based security model replacing authorities, permissions, object-level permissions, and groups; and database file migration tasks required for upgrading 12.7 databases.

Space Considerations

Flat FP indexes require more storage space in 16.0 than in earlier releases due to different architecture. Please plan for increased storage space requirements.

System Procedures Changes After Upgrading from 15.x

The way privileged system procedures run has changed with the enhanced role-based security.

In pre-16.0 versions of SAP Sybase IQ, a privileged system procedures run with the privileges of its owner, typically the dbo system role; this is called the SYSTEM PROCEDURE DEFINER model. In 16.0, privileged system procedures run with the privileges of the person executing it; this is called the SYSTEM PROCEDURE INVOKER model.

Note: This behavior change applies to SAP Sybase IQ privileged system procedures only, and not to user-defined stored procedures.

In the SYSTEM PROCEDURE DEFINER model, when you grant a user explicit EXECUTE object-level privilege on a system procedure, any privileges required to run any authorized tasks associated with the system procedure are automatically inherited from the owner (definer of the system procedure).

In the SYSTEM PROCEDURE INVOKER model, the EXECUTE object-level privilege for each system procedure is granted to the PUBLIC role. Since, by default, every user is a member of the PUBLIC role, every user automatically inherits the required EXECUTE object-level privilege. Any additional system privileges needed to run the system procedure are not inherited; you must now directly or indirectly grant them to the user.

This behavior change might create a loss of functionality on custom stored procedures and applications that explicitly grant EXECUTE object-level privilege on system procedures. For this reason, the default upgrade process for pre-16.0 database uses a combination of the two models. Privileged system procedures that existed in IQ versions earlier than 16.0 continue to run using the SYSTEM PROCEDURE DEFINER model, while any privileged system procedures introduced in 16.0 or later use the SYSTEM PROCEDURE INVOKER model.

You can override the default upgrade behavior so that all privileged system procedures for all versions use only the SYSTEM PROCEDURE INVOKER model if the potential loss of functionality is not a concern. If you are unsure of potential loss of functionality, upgrade you database to use the default behavior, then investigate. If you determine that it is not an issue, and you want all system procedures to run using the SYSTEM PROCEDURE INVOKER model, use the **ALTER DATABASE** statement to change the default security model. When you override the default upgrade behavior, the EXECUTE object-level privilege to run privileged system procedures is revoked from all users and roles and granted to the PUBLIC role.

To specify a security model, use the **CREATE DATABASE** statement, **ALTER DATABASE UPGRADE** statement, or the **Initialization** utility (**iqinit**).

Once a database is upgraded to SAP Sybase IQ 16.0, the default upgrade behavior for any subsequent upgrades (for example to 16.0 SP2, SP3, and so on) is to continue to use the current security model.

There is a small subset of pre-16.0 privileged system procedures that has always run with the privileges of the user running the procedure, rather than the owner of the procedure. In addition to EXECUTE object-level privilege on the system procedure, you must be granted additional system privileges that are specific to the system procedure.

When running privileged system procedures using the SYSTEM PROCEDURE DEFINER model, the DBO system role is typically the owner of the procedures. By default, the dbo system role is granted the SYS_AUTH_DBA_ROLE compatibility role. This ensures that the role is indirectly granted all privileges necessary to execute system procedures. Migrating the SYS_AUTH_DBA_ROLE compatibility role can result in the dbo system role losing the ability to execute privileged system procedures. See *Implications of Migrating Compatibility Roles on System Roles* for details.

Pre-16.0 Privileged System Procedures

A list of pre-16.0 privileged system procedures.

Privileged System Procedures Using the Combined Security Model

For these privileged system procedures, if the database is configured to use SYSTEM PROCEDURE DEFINER, you only need EXECUTE object-level privilege on the procedure to run it. If the database is configured to use SYSTEM PROCEDURE INVOKER, you also need the individual system privileges required by each procedure. Refer to the *Reference: Building Blocks, Tables, and Procedures* Guide for the system privileges require to run each system procedure.

- sa_audit_string
- sa_checkpoint_execute
- sa_disable_auditing_type
- sa_disk_free_space
- sa_enable_auditing_type
- sa_external_library_unload
- sa_flush_cache
- sa_list_external_library
- sa_server_option
- sa_procedure_profile
- sa_procedure_profile_summary
- sa_table_page_usage
- sa_validate

- sp_iq_reset_identity
- sp_iqaddlogin
- sp_iqbackupdetails
- sp_iqbackupsummary
- sp_iqcardinality_analysis
- sp_iqcheckdb
- sp_iqcheckoptions
- sp_iqclient_lookup
- sp_iqcolumn
- sp_iqcolumnuse
- sp_iqconnection
- sp_iqconstraint
- sp_iqcontext
- sp_iqconstraint
- sp_iqcontext
- sp_iqcursorinfo
- sp_iqdatatype
- sp_iqdbsize
- sp_iqdbspace
- sp_iqdbspaceinfo
- sp_iqdbspaceobjectinfo
- sp_iqdbstatistics
- sp_iqdroplogin
- sp_iqemptyfile
- sp_iqestdbspaces
- sp_iqestspace
- sp_iqevent
- sp_iqfile
- sp_iqhelp
- sp_iqindex
- sp_iqindex_alt
- sp_iqindexadvice
- sp_iqindexfragmentation
- sp_iqindexinfo
- sp_iqindexmetadata
- sp_iqindexsize
- sp_iqindexuse
- sp_iqlmconfig
- sp_iqlocks
- sp_iqmodifyadmin
- sp_iqmodifylogin
- sp_iqmpxcheckdqpconfig
- sp_iqmpxdumptlvlog
- sp_iqmpxfilestatus
- sp_iqmpxincconnpoolinfo
- sp_iqmpxincheartbeatinfo
- sp_iqcopyloginpolicy
- sp_iqmpxincconnpoolinfo
- sp_iqmpxincheartbeatinfo
- sp_iqmpxinfo
- sp_iqmpxversioninfo
- sp_iqobjectinfo
- sp_iqpkeys
- sp_iqprocedure
- sp_iqprocparm
- sp_iqrebuildindex
- sp_iqrename
- sp_iqrestoreaction
- sp_iqrowdensity
- sp_iqsetcompression
- sp_iqsharedtempdistrib
- sp_iqshowcompression
- sp_iqshowpsexe
- sp_iqspaceinfo
- sp_iqspaceused
- sp_iqstatistics
- sp_iqstatus
- sp_iqsysmon
- sp_iqtable
- sp_iqtablesize
- sp_iqtableuse
- sp_iqtransaction
- sp_iqunusedcolumn
- sp_iqunusedindex
- sp_iqunusedtable
- sp_iqversionuse
- sp_iqview
- sp_iqwho
- sp_iqworkmon

Privileged System Procedures Using Invoker Privileges

These pre-16.0 privileged system procedures run with the privileges of the user who is running the procedure, not the owner of the procedure, regardless of the security model setting. Therefore, in addition to the EXECUTE object-level privilege on the system procedure, (which is, by default, granted through membership in PUBLIC role), you must also be granted the additional system privileges required by the system procedure. Refer to the *Reference:*

Building Blocks, Tables, and Procedures Guide for the system privileges required to run each system procedure.

- sa_describe_shapefile
- sa_get_user_status
- sa_locks
- sa_performance_diagnostics
- sa_report_deadlocks
- sa_text_index_stats

Role-Based Security After Upgrading from 15.x

Role-based security replaces the authority-based security model as of SAP Sybase IQ 16.0.

What Happened to Authorities, Permissions, and Groups?

SAP Sybase IQ 16.0 introduces a role-based security model. Earlier versions used authorities, permissions, object-level permissions, and groups. The role-based security model uses roles, system privileges, object-level privileges, and user-extended roles.

Note: You can use an SAP Sybase IQ 16.0 or later server with an IQ database from an earlier version (pre-16.0). When you do, full backward compatibility is provided for that database, and its security model is not changed.

In pre-16.0, authorities are database-level permissions. For example, a user with BACKUP authority can back up the database. Some authorities bundle object-level permissions. For example, a user with PROFILE authority can perform application profiling and database tracing tasks, which involve using system procedures that are not otherwise available for use. You cannot create new authorities, alter the permissions they comprised, or drop them. You can grant administrative rights (WITH GRANT), but cannot limit the grant to administrator rights only.

In 16.0 or later, roles replace authorities, with the added benefit that you can create new roles, alter the system privileges granted, and drop them. Roles and privileges provide a more granular control when granting system privileges to users. You can also grant roles to users with administrative rights only. A user can then grant and revoke the role, but cannot exercise the underlying privileges.

In pre-16.0, permissions allowed you to create, modify, query, use, or delete database objects such as tables, views, and users.

In 16.0, system privileges replace permissions in functionality. Every privileged operation that you can perform on a database object now has a grantable system privilege. You can grant system privileges individually to users or to a role.

In 16.0, the meaning of *permissions* has changed. In pre-16.0 versions, it meant the a permission meant a grantable capability. Now, permission means the result of an evaluation of whether an operation can be performed. For example, you have permission to alter the table if you are the owner or you have the ALTER ANY TABLE system privilege.

In pre-16.0, groups were a collection of one or more users whose authorities and permissions were determined at the group level. A user was granted group status, and then other users were granted membership in that group.

In 16.0, the group paradigm is achieved using user-extended roles. If you have a user with a set of privileges that you want to grant to other users, you extend the user to become a user-extended role, and then grant that role to other users.

Upgrading a pre-16.0 database automatically converts your existing authority, permission, and group hierarchy into an equivalent role, privilege, and user-extended role hierarchy. Every pre-16.0 authority has a compatibility role. These roles are easily identifiable because their names start with SYS_AUTH. Compatibility roles contain the system privileges required to perform the same operations as the corresponding previous authority.

To take full advantage of the control and granularity of privileges available with role-based security, SAP recommends that, after migration, you review the compatibility role grants of each user and adjust membership and system privilege grants as necessary.

Authorities Become Compatibility Roles

When you upgrade to 16.0, users who are granted authorities in the earlier version are automatically granted an equivalent compatibility role. If the user previously had the ability to administer the authority, he or she has the ability to administer the compatibility role in 16.0.

The naming convention for each compatibility role retains the original authority name, but is prefaced with "SYS_AUTH_" and suffixed with "_ROLE". For example, the authority BACKUP becomes the role SYS_AUTH_BACKUP_ROLE, authority RESOURCE becomes role SYS_AUTH_RESOURCES_ROLE, and so on.

You cannot modify compatibility roles. However, you can migrate them to a user-defined role, and then modify them. When you migrate a compatibility role to a user-defined role, all users who are granted the compatibility role are granted the new user-defined role. Once each underlying system privilege is granted to at least one other role, you can drop the compatibility role. To restore compatibility roles, use the **CREATE ROLE** statement.

SQL statements are backward compatibility to support applications that grant or revoke authorities. However, the old syntax is deprecated and SAP recommends that you changed your applications to use the new SQL syntax for roles.

This table shows each authority and its equivalent compatibility role.

Authority	Compatibility Role	Description
BACKUP	SYS_AUTH_BACKUP_ROLE	Allows a user to back up databases and transaction logs with archive or image backups by using the **BACKUP DATABASE** statement or **dbbackup** utility.
DBA	SYS_AUTH_DBA_ROLE SYS_AUTH_SA_ROLE SYS_AUTH_SSO_ROLE	Allows users to perform all possible privileged operations. Users with the SYS_AUTH_DBA_ROLE role can create database objects and assign ownership of these objects to other user IDs, change table structures, create new user IDs, revoke permissions from users, back up the database, and so on. Of the possible privileged operations that the SYS_AUTH_DBA_ROLE compatibility role can perform, the SYS_AUTH_SA_ROLE compatibility role allows the user to perform all database administration-related activities, such as creating tables, and backing up data. Of the possible privileged operations that the SYS_AUTH_DBA_ROLE compatibility role can perform, the SYS_AUTH_SSO_ROLE compatibility role allows the user to perform the security and access-related administration activities, such as creating users, and granting privileges on objects.
PROFILE	SYS_AUTH_PROFILE_ROLE	Allows a user to perform profiling, tracing, and diagnostic operations.
READCLIENTFILE	SYS_AUTH_READCLIENTFILE_ROLE	Allows a user to read files on the client computer, for example when loading data from a file on a client computer.
READFILE	SYS_AUTH_READFILE_ROLE	Allows a user to use the OPENSTRING clause in a **SELECT** statement to read a file.
REMOTE DBA	SYS_RUN_REPLICATION_ROLE system role SYS_REPLICATION_ADMIN_ROLE system role	Allows a SQL Remote user to perform replication activities by using the **dbremote** and **dbmisync** utilities. It does not, however, allow administration of replication. The SYS_REPLICATION_ADMIN_ROLE system role is provided for replication administration.
RESOURCE	SYS_AUTH_RESOURCE_ROLE	Allows a user to create database objects, such as tables, views, stored procedures, and triggers.

Authority	Compatibility Role	Description
VALIDATE	SYS_AUTH_VALIDATE_ROLE	Allows a user to perform database, table, index, and checksum validation by using the **VALIDATE** statement or **dbvalid** utility.
WRITECLIENTFILE	SYS_AUTH_WRITECLIENTFILE_ROLE	Allows a user to write to files on a client computer, for example, when using the **UNLOAD TABLE** statement to write data to a client computer.
WRITEFILE	SYS_AUTH_WRITEFILE_ROLE	Allows a user to execute the **xp_write_file** system procedure.

In an authority-based security model, there was no way to limit the grant if a user did not need all of the permissions vested in an authority. This resulted in users often being granted more permissions than necessary, a potential security concern. The role-based security model addresses this concern by allowing system privileges to be granted at a granular level.

Since the migration process ensures that all existing privileges are preserved at the user and group level, SAP recommends that, after migration, you review the compatibility role grants for each user and adjust membership as necessary.

Permissions Become Privileges

SAP Sybase IQ versions earlier than 16.0 used object-level permissions such as ALTER and SELECT for tables and views. These permissions are now called object-level privileges.

When you upgrade your database, users granted permissions are automatically granted the equivalent object-level privileges.

SQL statements that grant or revoke object-level permissions/privileges still work.

In addition to object-level privileges, there are now grantable system privileges for every authorized operation.

Groups Become Roles

When you upgrade an IQ database earlier than 16.0, each group is converted to an equivalent user-extended role of the same name. Members of the original group are automatically granted the new role and all of its underlying system privileges. Authorities and object-level permissions granted to the original group are converted to equivalent compatibility roles and system privileges and are granted to the user-extended role.

If an authority was inheritable, grantees of the new user-extended role inherit the underlying system privileges of the equivalent compatibility role. If the authority was not inheritable, the grantees of the user-extended role do not inherit the underlying system privileges of the equivalent compatibility role. If the legacy group had a password, only the extended user of the user-extended role inherits the underlying system privileges of the compatibility role that is not inheritable.

System groups become system roles with the same name.

Group/System Role	Description
dbo	This role owns many system stored procedures, views, and tables.
diagnostics	This role owns the diagnostic tables and views, and can perform operations on them.
PUBLIC	This role has SELECT permission on the system tables. Any new user ID is automatically granted the PUBLIC role.
ra_systabgroup	This role allows users to perform replication server functionality.
SYS	This role owns the system tables and views (IQ catalog) for the database, and can perform operations on them.
SYS_SPATIAL_ADMIN_ROLE	This role allows users to create, alter, or drop spatial objects.

Change to Concept of a Super-User (DBA Authority)

In SAP Sybase IQ versions earlier than 16.0, the DBA user was often considered a super-user by virtue of being granted the DBA authority. The DBA user continues to exist in 16.0, however, the concept has changed.

In SAP Sybase IQ versions earlier than 16.0, you created a super-user by granting a user DBA authority. Users with DBA authority can perform any privileged task in the system. When you upgrade your database to 16.0, any users granted DBA authority are granted the equivalent SYS_AUTH_DBA_ROLE compatibility role and become a super-user. After migration, the super-user itself must be managed to preserve its super-user abilities.

When you create a new role without specify an administrator, users with the MANAGE ROLES system privilege (global administrators) can administer the role. Since MANAGE ROLES is one of the system privileges granted to the SYS_AUTH_DBA_ROLE compatibility role, super-users can administer new roles.

However, if you create a new role and assign administrators, administration is then limited to only those administrators; global administrators, and thus super-users, cannot administer the role. You must explicitly assign the super-user as an administrator of the role.

You can migrate the SYS_AUTH_DBA_ROLE compatibility role to a user-defined role, and eventually drop it. Since the ability of the super-user to perform any privileged task is dependent on the SYS_AUTH_DBA_ROLE compatibility role, before you drop the SYS_AUTH_DBA_ROLE compatibility role, you must grant the super-user membership in the new user-defined role or grant each of the underlying system privileges directly or indirectly to the super-user.

Changes to the GRANT Statement Syntax

The **GRANT** syntax for authorities, permissions, and groups is supported, but deprecated. If you have applications that use the pre-16.0 GRANT statement syntax, modify them to use the new syntax for roles and privileges.

In SAP Sybase IQ versions earlier than 16.0, DBA, REMOTE DBA, RESOURCE, and VALIDATE authorities were not inheritable. When you upgrade your database to 16.0, the WITH NO SYSTEM PRIVILEGE INHERITANCE clause ensures that inheritance behavior remains consistent with earlier releases.

Users granted DBA and REMOTE DBA authorities could grant them to others. In 16.0, the WITH ADMIN clause is introduced to ensure that administration rights behavior remains consistent with previous releases.

Table 1. Non-Inheritable Authorities

Pre-16.0 Syntax	New Syntax
GRANT DBA TO *<grantee>[,...]*	**GRANT ROLE SYS_AUTH_DBA_ROLE TO** *<grantee> [,...]* **WITH ADMIN OPTION** **WITH NO SYSTEM PRIVILEGE INHERITANCE**
GRANT REMOTE DBA TO *<grantee>[,...]*	**GRANT ROLE SYS_RUN_REPLICATION_ROLE TO** *<grantee> [,...]* **WITH NO ADMIN OPTION** **WITH NO SYSTEM PRIVILEGE INHERITANCE**
GRANT BACKUP TO *<grantee>[,...]*	**GRANT ROLE SYS_AUTH_BACKUP_ROLE TO** *<grantee> [,...]* **WITH NO SYSTEM PRIVILEGE INHERITANCE**
GRANT RESOURCE TO *<grantee>[,...]*	**GRANT ROLE SYS_AUTH_RESOURCE_ROLE TO** *<grantee> [,...]* **WITH NO SYSTEM PRIVILEGE INHERITANCE**
GRANT VALIDATE TO *<grantee>[,...]*	**GRANT ROLE SYS_AUTH_VALIDATE_ROLE TO** *<grantee> [,...]* **WITH NO SYSTEM PRIVILEGE INHERITANCE**

Table 2. Inheritable Authorities

Pre-16.0 Syntax	New Syntax
GRANT Multiplex Admin TO *<grantee> [,...]*	**GRANT ROLE SYS_AUTH_MULTIPLEX_ADMIN_ROLE TO** *<grantee> [,...]*
GRANT Operator TO *<grantee> [,...]*	**GRANT ROLE SYS_AUTH_OPERATOR_ROLE TO** *<grantee> [,...]*
GRANT Perms Admin TO *<grantee> [,...]*	**GRANT ROLE SYS_AUTH_PERMS_ADMIN_ROLE TO** *<grantee> [,...]*
GRANT PROFILE TO *<grantee> [,...]*	**GRANT ROLE SYS_AUTH_PROFILE_ROLE TO** *<grantee> [,...]*
GRANT READCLIENTFILE TO *<grantee> [,...]*	**GRANT ROLE SYS_AUTH_READCLIENTFILE_ROLE TO** *<grantee> [,...]*
GRANT READFILE TO *<grantee> [,...]*	**GRANT ROLE SYS_AUTH_READFILE_ROLE TO** *<grantee> [,...]*
GRANT Space Admin TO *<grantee> [,...]*	**GRANT ROLE SYS_AUTH_SPACE_ADMIN_ROLE TO** *<grantee> [,...]*
GRANT Spatial Admin TO *<grantee> [,...]*	**GRANT ROLE SYS_AUTH_SPATIAL_ADMIN_ROLE TO** *<grantee> [,...]*
GRANT WRITECLIENTFILE TO *<grantee> [,...]*	**GRANT ROLE SYS_AUTH_WRITECLIENTFILE_ROLE TO** *<grantee> [,...]*
GRANT WRITEFILE TO *<grantee> [,...]*	**GRANT ROLE SYS_AUTH_WRITEFILE_ROLE TO** *<grantee> [,...]*
GRANT CONNECT TO *<username>* [**IDENTIFIED BY** *<pwd>*]	No change
GRANT GROUP TO *<user>*	**CREATE OR REPLACE** *<rolename>* **FOR USER** *<user>*
GRANT MEMBERSHIP IN GROUP *<groupname>[,...]* **TO** *<grantee>[,...]*	**GRANT ROLE** *<groupname>[,...]* **TO** *<grantee>[,...]*
GRANT PUBLISH TO *<grantee>*	No change; however, you can also set the new PUBLIC option, db_publisher: **SET OPTION PUBLIC.db_publisher=***<grantee_id>*

Pre-16.0 Syntax	New Syntax
GRANT *<permission>[,...]* **ON** *[owner.]object-name* **TO** *<grantee>[,...]* **[WITH GRANT OPTION]** *<permission>*: ALL [PRIVILEGES] | ALTER | DELETE | INSERT | REFERENCES [(column-name, ...)] | SELECT [(column-name, ...)] | UPDATE [(column-name, ...)]	No change
GRANT EXECUTE ON *[owner.]{ procedure-name* | *user-defined-function* } **TO** *<grantee>[,...]*	No change
GRANT INTEGRATED LOGIN TO *<user-profile-name>[,...]* **AS USER** *<user>*	No change
GRANT KERBEROS LOGIN **TO** *client-Kerberos-principal [, …]* **AS USER** *<user>*	No change
GRANT CREATE ON *<dbspacename> [,...]* **TO** *<grantee> [,...]*	No change

Changes to the REVOKE Statement Syntax

The **REVOKE** syntax for authorities, permissions, and groups is supported but deprecated. If you have applications that use the pre-16.0 REVOKE statement syntax, modify them to use the new syntax for roles and privileges.

Pre-16.0 Syntax	New Syntax
REVOKE CONNECT FROM *<user>*	No change

Pre-16.0 Syntax	New Syntax
REVOKE GROUP FROM *<user>*	**DROP** *<rolename>* **FROM USER** *<user>* **WITH REVOKE**
REVOKE MEMBERSHIP IN GROUP *<groupname> [,...]* **FROM** *<grantee> [,...]*	**REVOKE ROLE** *<groupname>[,...]* **FROM** *<grantee> [,...]*
REVOKE *<authority>[,...]* **FROM** *<grantee> [,...]* *<authority>*: BACKUP |DBA |Multiplex Admin |Operator |Perms Admin |PROFILE |READCLIENTFILE |READFILE |REMOTE DBA |RESOURCE | ALL |Space Admin |Spatial Admin |User Admin |VALIDATE |WRITECLIENTFILE |WRITEFILE	**REVOKE** *<rolename>[,...]* **FROM** *<grantee> [,...]* *<rolename>*: SYS_AUTH_BACKUP_ROLE |SYS_AUTH_DBA_ROLE |SYS_AUTH_MULTIPLEX_ADMIN_ROLE |SYS_AUTH_OPERATOR_ROLE |SYS_AUTH_PERMS_ADMIN_ROLE |SYS_AUTH_PROFILE_ROLE |SYS_READCLIENTFILE_ROLE |SYS_AUTH_READFILE_ROLE |SYS_RUN_REPLICATION_ROLE |SYS_AUTH_RESOURCE_ROLE |SYS_AUTH_SPACE_ADMIN_ROLE |SYS_AUTH_SPATIAL_ADMIN_ROLE |SYS_AUTH_USER_ADMIN_ROLE |SYS_AUTH_VALIDATE_ROLE |SYS_AUTH_WRITECLIENTFILE_ROLE |SYS_AUTH_WRITEFILE_ROLE
REVOKE PUBLISH FROM *grantee*	No change; however, you can also set the new PUBLIC option, db_publisher: **SET OPTION PUBLIC.db_publisher=grantee**

Pre-16.0 Syntax	New Syntax
REVOKE *<permission>[,...]* **ON** *[owner.]object-name* **FROM** *<grantee>[,...]* *<permission>*: ALL [PRIVILEGES] | ALTER | DELETE | INSERT | REFERENCES [(column-name, ...)] | SELECT [(column-name, ...)] | UPDATE [(column-name, ...)]	No change, except to naming convention. Object-level permissions are now object-level privileges.

REVOKE EXECUTE ON [*owner.]{ procedure-name | user-defined-function* }
FROM *<grantee> [,...]*
	No change

REVOKE INTEGRATED LOGIN FROM *<user>*	No change
REVOKE KERBEROS LOGIN FROM *<user> [,...]* **AS USER** *<user>*	No change
REVOKE CREATE ON *<dbspacename> [,...]* **FROM** *<grantee> [,...]*	No change

Changes to REMOTE DBA

The REMOTE DBA authority is replaced by the SYS_RUN_REPLICATION_ROLE system role. This role allows user to administer replication.

The GRANT REMOTE DBA statement syntax is still supported but deprecated. Change your applications to grant this new role.

Changes in Inheritance Behavior

In SAP Sybase IQ versions earlier than 16.0, when you granted the DBA, REMOTE DBA, BACKUP, RESOURCE, and VALIDATE authorities to a group, the underlying permissions were not inherited by members of the group.

In 16.0, the default behavior when granting SYS_AUTH_DBA_ROLE, SYS_RUN_REPLICATION_ROLE, SYS_AUTH_BACKUP_ROLE, SYS_AUTH_RESOURCE_ROLE, and SYS_AUTH_VALIDATE_ROLE (formerly DBA, REMOTE DBA, BACKUP, RESOURCE, and VALIDATE) to a user-defined role allows members of the user-defined role to inherit the underlying system privileges of the role. The WITH NO SYSTEM INHERITANCE clause retains the non-inheritance behavior.

To retain the non-inheritance behavior of these roles after upgrading, include the WITH NO SYSTEM PRIVILEGE INHERITANCE clause in the **GRANT ROLE** statement, if you have applications that you are changing to use the new **GRANT** syntax, and want to preserve the non-inheritance behavior.

Note: The WITH NO SYSTEM PRIVILEGE INHERITANCE clause is supported only with these specific roles; any other use results in an error.

For example, you grant `User1` the ALTER ANY OBJECT system privilege. You extend `User1` to become a role and grant `User1` to `User2`. You want to grant the SYS_AUTH_DBA_ROLE role to `User1`, but you do not want `User2` to inherit the system privileges vested to SYS_AUTH_DBA_ROLE. Execute:

```
GRANT ROLE SYS_AUTH_DBA_ROLE TO User1 WITH NO SYSTEM PRIVILEGE
INHERITANCE;
```

Changes in Administering the Database Publisher

In 16.0, the PUBLISH authority is replaced by the **PUBLIC.db_publisher** database option. You change the publisher by changing the database option.

In IQ versions earlier than 16.0, the database publisher was controlled by using the **GRANT PUBLISH** and **REVOKE PUBLISH** statements. You determine the current publisher by querying the CURRENT PUBLISHER special value. The **GRANT PUBLISH** and **REVOKE PUBLISH** syntax is still supported, but deprecated.

You can also still query the CURRENT PUBLISHER to determine the current publisher.

The SET ANY SYSTEM OPTION system privilege is required to set **PUBLIC.db_publisher**.

Compatibility Roles

Compatibility roles exist for backward compatibility with versions of SAP Sybase IQ earlier than 16.0. that support authority-based security.

You can grant, revoke, and under specific conditions, delete compatibility roles. You cannot modify any of the underlying system privileges. However, you can migrate compatibility roles to user-defined roles, and then modify the underlying system privileges. When you migrate a compatibility role, all grantees of the compatibility role are automatically granted the user-defined role.

Use the compatibility roles SYS_AUTH_SA_ROLE and SYS_AUTH_SSO_ROLE to administer and grant all individual system privileges in a new database. The union of the system privileges of these two roles is granted to the compatibility role

SYS_AUTH_DBA_ROLE. By default, SYS_AUTH_DBA_ROLE is granted to the DBA user with administrative privileges. Thus, all system privileges are initially granted to the DBA user.

You can grant users to, and revoke users from, compatibility roles.

Grant Compatibility Roles

Granting a compatibility role is semantically equivalent to granting each of its underlying system privileges and roles.

Granting SYS_AUTH_SA_ROLE

Allows users to perform authorized tasks pertaining to data and system administration responsibilities.

Prerequisites

Administrative privilege over SYS_AUTH_SA_ROLE role.

Task

You can grant this role with or without administrative rights. When granted with administrative rights, you can manage (grant and revoke) the role, as well as use any of the underlying system privileges. When granted with administrative rights only, you can manage the role, but not use its underlying system privileges. Finally, when granted with no administrative rights, you can only use its underlying system privileges.
To grant the SYS_AUTH_SA_ROLE role, execute one of these statements:

Administrative Option	Statement
With full administrative rights	**GRANT ROLE SYS_AUTH_SA_ROLE TO** *grantee [,...]* **WITH ADMIN OPTION**
With administrative rights only	**GRANT ROLE SYS_AUTH_SA_ROLE TO** *grantee [,...]* **WITH ADMIN ONLY OPTION**
With no administrative rights	**GRANT ROLE SYS_AUTH_SA_ROLE TO** *grantee [,...]* **WITH NO ADMIN OPTION**

System Privileges Granted to SYS_AUTH_SA_ROLE

Each system privilege is granted with full administrative rights.

- ACCESS SERVER LS
- ALTER ANY INDEX
- ALTER ANY MATERIALIZED VIEW
- ALTER ANY OBJECT

- ALTER ANY PROCEDURE
- ALTER ANY SEQUENCE
- ALTER ANY TEXT CONFIGURATION
- ALTER ANY TABLE
- ALTER ANY TRIGGER
- ALTER ANY VIEW
- ALTER DATABASE
- ALTER DATATYPE
- BACKUP DATABASE
- CHECKPOINT
- COMMENT ANY OBJECT
- CREATE ANY INDEX
- CREATE ANY MATERIALIZED VIEW
- CREATE ANY OBJECT
- CREATE ANY PROCEDURE
- CREATE ANY SEQUENCE
- CREATE ANY TABLE
- CREATE ANY TEXT CONFIGURATION
- CREATE ANY TRIGGER
- CREATE ANY VIEW
- CREATE DATATYPE
- CREATE EXTERNAL REFERENCE
- CREATE MATERIALIZED VIEW
- CREATE MESSAGE
- CREATE PROCEDURE
- CREATE PROXY TABLE
- CREATE TABLE
- CREATE TEXT CONFIGURATION
- CREATE VIEW
- DEBUG ANY PROCEDURE
- DELETE ANY TABLE
- DROP ANY INDEX
- DROP ANY MATERIALIZED VIEW
- DROP ANY OBJECT
- DROP ANY PROCEDURE
- DROP ANY SEQUENCE
- DROP ANY TABLE
- DROP ANY TEXT CONFIGURATION
- DROP ANY VIEW

- DROP DATATYPE
- DROP MESSAGE
- EXECUTE ANY PROCEDURE
- INSERT ANY TABLE
- LOAD ANY TABLE
- MANAGE ANY DBSPACE
- MANAGE ANY EVENT
- MANAGE ANY EXTERNAL ENVIRONMENT
- MANAGE ANY EXTERNAL OBJECT
- MANAGE ANY MIRROR SERVER
- MANAGE ANY SPATIAL OBJECT
- MANAGE ANY STATISTICS
- MANAGE ANY WEB SERVICE
- MANAGE MULTIPLEX
- MANAGE PROFILING
- MANAGE REPLICATION
- MONITOR
- READ CLIENT FILE
- READ FILE
- REORGANIZE ANY OBJECT
- SELECT ANY TABLE
- SERVER OPERATOR
- SET ANY PUBLIC OPTION
- SET ANY SYSTEM OPTION
- SET ANY USER DEFINED OPTION
- TRUNCATE ANY TABLE
- UPDATE ANY TABLE
- UPGRADE ROLE
- USE ANY SEQUENCE
- VALIDATE ANY OBJECT
- WRITE CLIENT FILE
- WRITE FILE

Granting SYS_AUTH_SSO_ROLE

Allows users to perform authorized tasks pertaining to security and access control responsibilities.

Prerequisites

Administrative privilege over SYS_AUTH_SSO_ROLE role.

Task

You can grant this role with or without administrative rights. When granted with administrative rights, you can manage (grant and revoke) the role, as well as use any of the underlying system privileges. When granted with administrative rights only, you can manage the role, but not use its underlying system privileges. Finally, when granted with no administrative rights, you can only use its underlying system privileges.

To grant the role, execute one of these statements:

Administrative Option	Statement
With full administrative rights	**GRANT ROLE SYS_AUTH_SSO_ROLE TO** *grantee [,...]* **WITH ADMIN OPTION**
With administrative rights only	**GRANT ROLE SYS_AUTH_SSO_ROLE TO** *grantee [,...]* **WITH ADMIN ONLY OPTION**
With no administrative rights	**GRANT ROLE SYS_AUTH_SSO_ROLE TO** *grantee [,...]* **WITH NO ADMIN OPTION**

System Privileges Granted to SYS_AUTH_SSO_ROLE

Each system privilege is granted with full administrative rights only unless noted.

- ALTER ANY OBJECT OWNER
- ANY USER
- CHANGE PASSWORD
- DROP CONNECTION
- MANAGE ANY OBJECT PRIVILEGES
- MANAGE ANY LDAP SERVER
- MANAGE ANY LOGIN POLICY
- MANAGE ANY USER
- MANAGE AUDITING
- MANAGE ROLES
- SET ANY SECURITY OPTION
- SET USER (granted with the administrative rights only)

Granting SYS_AUTH_DBA_ROLE

Allows users to perform all authorized tasks.

Prerequisites

Administrative privilege over SYS_AUTH_DBA_ROLE role.

Task

This role indirectly grants all compatibility roles, and some system roles, to a user. It is the union of the underlying system privileges of each of these roles that makes the SYS_AUTH_DBA_ROLE role the "super" role.

You can grant this role with or without administrative rights. When granted with administrative rights, you can manage (grant and revoke) the role, as well as use any of the underlying system privileges. When granted with administrative rights only, you can manage the role, but not use its underlying system privileges. Finally, when granted with no administrative rights, you can only use its underlying system privileges.

Note: When migrating from SAP Sybase IQ 15.4 and earlier, the concept of inheritance of the underlying system privileges of this system role represents a change in behavior in SAP Sybase IQ 16.0 or later. For SAP Sybase IQ 15.4 and earlier behavior, use the WITH NO SYSTEM PRIVILEGE INHERITANCE clause.

The WITH ADMIN ONLY OPTION clause is invalid when using the WITH NO SYSTEM PRIVILEGE INHERITANCE clause. The WITH NO ADMIN OPTION clause is valid, but not required, as it is semantically equivalent to the WITH NO SYSTEM PRIVILEGE INHERITANCE clause.

To grant the SYS_AUTH_DBA_ROLE role, execute one of these statements:

Administrative Option	Statement
With full administrative rights	**GRANT ROLE SYS_AUTH_DBA_ROLE TO** *grantee [,...]* **WITH ADMIN OPTION**
With administrative rights only	**GRANT ROLE SYS_AUTH_DBA_ROLE TO** *grantee [,...]* **WITH ADMIN ONLY OPTION**
With no administrative rights	**GRANT ROLE SYS_AUTH_DBA_ROLE TO** *grantee [,...]* **WITH NO ADMIN OPTION**
With full administrative rights, **but no system privilege inheritance**	**GRANT ROLE SYS_AUTH_REMOTE_DBA_ROLE** TO *user_ID* **WITH ADMIN OPTION** **WITH NO SYSTEM PRIVILEGE INHERITANCE**

Roles Granted to SYS_AUTH_DBA_ROLE

System and compatibility roles granted to the SYS_AUTH_DBA_ROLE role.

These compatibility roles are granted with full administrative rights:

- SYS_AUTH_SA_ROLE

- SYS_AUTH_SSO_ROLE

These compatibility roles are granted with administrative rights only:

- SYS_AUTH_RESOURCE_ROLE
- SYS_AUTH_BACKUP_ROLE
- SYS_AUTH_VALIDATE_ROLE
- SYS_AUTH_READFILE_ROLE
- SYS_AUTH_PROFILE_ROLE
- SYS_AUTH_READCLIENTFILE_ROLE
- SYS_AUTH_WRITECLIENTFILE_ROLE
- SYS_AUTH_WRITEFILE_ROLE
- SYS_AUTH_USER_ADMIN_ROLE
- SYS_AUTH_SPACE_ADMIN_ROLE
- SYS_AUTH_MULTIPLEX_ADMIN_ROLE
- SYS_AUTH_OPERATOR_ROLE
- SYS_AUTH_PERMS_ADMIN_ROLE

These system roles are granted with administrative rights only:

- SYS_SPATIAL_ADMIN_ROLE
- diagnostics
- rs_systabgroup
- SYS
- DBO
- PUBLIC

System Privileges Granted to SYS_AUTH_DBA_ROLE

Through the granting of all compatibility roles and select system roles, system privileges are indirectly granted to the SYS_AUTH_DBA_ROLE role. The underlying system privileges of the SYS_AUTH_SA_ROLE and SYS_AUTH_SSO_ROLE roles are indirectly granted with full administrative rights. All other compatibility roles and system roles are indirectly granted with administrative rights only.

- ACCESS SERVER LS
- ALTER ANY INDEX
- ALTER ANY MATERIALIZED VIEW
- ALTER ANY OBJECT
- ALTER ANY OBJECT OWNER
- ALTER ANY PROCEDURE
- ALTER ANY SEQUENCE
- ALTER ANY TABLE
- ALTER ANY TEXT CONFIGURATION

- ALTER ANY TRIGGER
- ALTER ANY VIEW
- ALTER DATABASE
- ALTER DATATYPE
- BACKUP DATABASE
- CHANGE PASSWORD
- CHECKPOINT
- COMMENT ANY OBJECT
- CREATE ANY INDEX
- CREATE ANY MATERIALIZED VIEW
- CREATE ANY OBJECT
- CREATE ANY PROCEDURE
- CREATE ANY SEQUENCE
- CREATE ANY TABLE
- CREATE ANY TEXT CONFIGURATION
- CREATE ANY TRIGGER
- CREATE ANY VIEW
- CREATE DATATYPE
- CREATE EXTERNAL REFERENCE
- CREATE MATERIALIZED VIEW
- CREATE MESSAGE
- CREATE PROCEDURE
- CREATE PROXY TABLE
- CREATE TABLE
- CREATE TEXT CONFIGURATION
- CREATE VIEW
- DEBUG ANY PROCEDURE
- DELETE ANY TABLE
- DROP ANY INDEX
- DROP ANY MATERIALIZED VIEW
- DROP ANY OBJECT
- DROP ANY PROCEDURE
- DROP ANY SEQUENCE
- DROP ANY TABLE
- DROP ANY TEXT CONFIGURATION
- DROP ANY VIEW
- DROP CONNECTION
- DROP DATATYPE
- DROP MESSAGE

- EXECUTE ANY PROCEDURE
- LOAD ANY TABLE
- INSERT ANY TABLE
- MANAGE ANY DBSPACE
- MANAGE ANY EVENT
- MANAGE ANY EXTERNAL ENVIRONMENT
- MANAGE ANY EXTERNAL OBJECT
- MANAGE ANY LDAP SERVER
- MANAGE ANY LOGIN POLICY
- MANAGE ANY MIRROR SERVER
- MANAGE ANY OBJECT PRIVILEGES
- MANAGE ANY SPATIAL OBJECT
- MANAGE ANY STATISTICS
- MANAGE ANY USER
- MANAGE ANY WEB SERVICE
- MANAGE AUDITING
- MANAGE MULTIPLEX
- MANAGE PROFILING
- MANAGE REPLICATION
- MANAGE ROLES
- MONITOR
- READ CLIENT FILE
- READ FILE
- REORGANIZE ANY OBJECT
- SELECT ANY TABLE
- SERVER OPERATOR
- SET ANY PUBLIC OPTION
- SET ANY SECURITY OPTION
- SET ANY SYSTEM OPTION
- SET ANY USER DEFINED OPTION
- SET USER (granted with administrative rights only)
- TRUNCATE ANY TABLE
- UPDATE ANY TABLE
- UPGRADE ROLE
- USE ANY SEQUENCE
- VALIDATE ANY OBJECT
- WRITE CLIENT FILE
- WRITE FILE

Granting SYS_AUTH_BACKUP_ROLE

Allows users to perform all backups.

Prerequisites

Administrative privilege over SYS_AUTH_BACKUP_ROLE.

Task

You can grant this role with or without administrative rights. When granted with administrative rights, you can manage (grant and revoke) the role, as well as use any of the underlying system privileges. When granted with administrative rights only, you can manage the role, but not use its underlying system privileges. Finally, when granted with no administrative rights, you can only use its underlying system privileges.

Note: When migrating from SAP Sybase IQ 15.4 and earlier, the concept of inheritance of the underlying system privileges of this system role represents a change in behavior in SAP Sybase IQ 16.0 or later. For SAP Sybase IQ 15.4 and earlier behavior, use the WITH NO SYSTEM PRIVILEGE INHERITANCE clause.

The WITH ADMIN ONLY OPTION and WITH ADMIN OPTION clauses are invalid when using the WITH NO SYSTEM PRIVILEGE INHERITANCE clause. The WITH NO ADMIN OPTION clause is valid, but not required, as it is semantically equivalent to the WITH NO SYSTEM PRIVILEGE INHERITANCE clause.

To grant the SYS_AUTH_BACKUP_ROLE role, execute one of the following statements:

Administrative Option	Statement
With full administrative rights	**GRANT ROLE SYS_AUTH_BACKUP_ROLE** TO *user_ID* **WITH ADMIN OPTION**
With administrative rights only	**GRANT ROLE SYS_AUTH_BACKUP_ROLE** TO *user_ID* **WITH ADMIN ONLY OPTION**
With no administrative rights	**GRANT ROLE SYS_AUTH_BACKUP_ROLE** TO *user_ID* **WITH NO ADMIN OPTION**
With no system privilege inheritance	**GRANT ROLE SYS_AUTH_BACKUP_ROLE** TO *user_ID* **WITH NO SYSTEM PRIVILEGE INHERITANCE**

Example:

This example grants the SYS_AUTH_BACKUP_ROLE to `Mary` with administrative rights and inheritance of the underlying system privileges of the role.

```
GRANT ROLE SYS_AUTH_BACKUP_ROLE TO Mary WITH ADMIN OPTION
```

This example grants the SYS_AUTH_BACKUP_ROLE to `Joe` with neither administrative rights nor inheritance of the roles underlying system privileges.

```
GRANT ROLE SYS_AUTH_BACKUP_ROLE TO Joe
WITH NO SYSTEM PRIVILEGE INHERITANCE
```

System Privileges Granted to SYS_AUTH_BACKUP_ROLE

The BACKUP DATABASE system privilege is granted with no administrative rights.

Granting SYS_AUTH_MULTIPLEX_ADMIN_ROLE

Allows users to perform authorized tasks to manage multiplex.

Prerequisites

Administrative privilege over SYS_AUTH_MULTIPLEX_ADMIN_ROLE.

Task

You can grant this role with or without administrative rights. When granted with administrative rights, you can manage (grant and revoke) the role, as well as use any of the underlying system privileges. When granted with administrative rights only, you can manage the role, but not use its underlying system privileges. Finally, when granted with no administrative rights, you can only use its underlying system privileges.

To grant the SYS_AUTH_MULTIPLEX_ADMIN_ROLE role, execute one of the following statements:

Administrative Option	Statement
With full administrative rights	**GRANT ROLE SYS_AUTH_MULTIPLEX_ADMIN_ROLE** TO *user_ID* **WITH ADMIN OPTION**
With administrative rights only	**GRANT ROLE SYS_AUTH_MULTIPLEX_ADMIN_ROLE** TO *user_ID* **WITH ADMIN ONLY OPTION**
With no administrative rights	**GRANT ROLE SYS_AUTH_MULTIPLEX_ADMIN_ROLE** TO *user_ID* **WITH NO ADMIN OPTION**

Example:

This example grants the SYS_AUTH_MULTIPLEX_ADMIN_ROLE to `Mary`, with no administrative rights.

```
GRANT ROLE SYS_AUTH_MULTIPLEX_ADMIN_ROLE TO Mary WITH NO ADMIN OPTION
```

System Privileges Granted to SYS_AUTH_MULTIPLEX_ADMIN_ROLE

The ACCESS SERVER LS and MANAGE MULTIPLEX system privileges are granted with the no administrative rights.

Granting SYS_AUTH_OPERATOR_ROLE

Allows users to checkpoint databases, drop connections (including those for users with SYS_AUTH_DBA_ROLE), back up databases, and monitor the system.

Prerequisites

Administrative privilege over SYS_AUTH_OPERATOR_ROLE.

Task

You can grant this role with or without administrative rights. When granted with administrative rights, you can manage (grant and revoke) the role, as well as use any of the underlying system privileges. When granted with administrative rights only, you can manage the role, but not use its underlying system privileges. Finally, when granted with no administrative rights, you can only use its underlying system privileges.

Note: When migrating from SAP Sybase IQ 15.4 and earlier, the concept of inheritance of the underlying system privileges of this system role represents a change in behavior in SAP Sybase IQ 16.0 or later. For SAP Sybase IQ 15.4 and earlier behavior, use the WITH NO SYSTEM PRIVILEGE INHERITANCE clause.

The WITH ADMIN ONLY OPTION and WITH ADMIN OPTION clauses are invalid when using the WITH NO SYSTEM PRIVILEGE INHERITANCE clause. The WITH NO ADMIN OPTION clause is valid, but not required, as it is semantically equivalent to the WITH NO SYSTEM PRIVILEGE INHERITANCE clause.

To grant the SYS_AUTH_OPERATOR_ROLE role, execute one of the following statements:

Administrative Option	Statement
With full administrative rights	**GRANT ROLE SYS_AUTH_OPERATOR_ROLE** TO *user_ID* **WITH ADMIN OPTION**
With administrative rights only	**GRANT ROLE SYS_AUTH_OPERATOR_ROLE** TO *user_ID* **WITH ADMIN ONLY OPTION**
With no administrative rights	**GRANT ROLE SYS_AUTH_OPERATOR_ROLE** TO *user_ID* **WITH NO ADMIN OPTION**

Administrative Option	Statement
With no system privilege inheritance	**GRANT ROLE SYS_AUTH_OPERATOR_ROLE** TO *user_ID* **WITH NO SYSTEM PRIVILEGE INHERITANCE**

Example:

This example grants the SYS_AUTH_OPERATOR_ROLE to `Mary` with administrative rights and inheritance of the underlying system privileges of the role.

```
GRANT ROLE SYS_AUTH_OPERATOR_ROLE TO Mary WITH ADMIN OPTION
```

This example grants the SYS_AUTH_OPERATOR_ROLE to `Joe` with neither administrative rights or inheritance of the underlying system privileges of the role.

```
GRANT ROLE SYS_AUTH_OPERATOR_ROLE TO Joe
WITH NO SYSTEM PRIVILEGE INHERITANCE
```

System Privileges Granted to SYS_AUTH_OPERATOR_ROLE

Each system privilege is granted with no administrative rights.

- ACCESS SERVER LS
- BACKUP DATABASE
- CHECKPOINT
- DROP CONNECTION
- MONITOR

Granting SYS_AUTH_PERMS_ADMIN_ROLE

Allows users to manage data privileges, groups, authorities, and passwords.

Prerequisites

Administrative privilege over SYS_AUTH_PERMS_ADMIN_ROLE.

Task

You can grant this role with or without administrative rights. When granted with administrative rights, you can manage (grant and revoke) the role, as well as use any of the underlying system privileges. When granted with administrative rights only, you can manage the role, but not use its underlying system privileges. Finally, when granted with no administrative rights, you can only use its underlying system privileges.

To grant the SYS_AUTH_PERMS_ADMIN_ROLE role, execute one of the following statements:

Administrative Option	Statement
With full administrative rights	**GRANT ROLE SYS_AUTH_PERMS_ADMIN_ROLE** TO *user_ID* **WITH ADMIN OPTION**
With administrative rights only	**GRANT ROLE SYS_AUTH_PERMS_ADMIN_ROLE** TO *user_ID* **WITH ADMIN ONLY OPTION**
With no administrative rights	**GRANT ROLE SYS_AUTH_PERMS_ADMIN_ROLE** TO *user_ID* **WITH NO ADMIN OPTION**

Example:

This example grants the SYS_AUTH_PERMS_ADMIN_ROLE to `Mary`, with only administrative rights.

```
GRANT ROLE SYS_AUTH_PERMS_ADMIN_ROLE TO Mary WITH ADMIN ONLY OPTION
```

Roles Granted to SYS_AUTH_PERMS_ADMIN_ROLE

Compatibility roles granted with full administrative rights.

- SYS_AUTH_BACKUP_ROLE
- SYS_AUTH_OPERATOR_ROLE
- SYS_AUTH_USER_ADMIN_ROLE
- SYS_AUTH_SPACE_ADMIN_ROLE
- SYS_AUTH_MULTIPLEX_ADMIN_ROLE
- SYS_AUTH_RESOURCE_ROLE
- SYS_AUTH_VALIDATE_ROLE
- SYS_AUTH_PROFILE_ROLE
- SYS_AUTH_READFILE_ROLE
- SYS_AUTH_READCLIENTFILE_ROLE
- SYS_AUTH_WRITEFILE_ROLE
- SYS_AUTH_WRITECLIENTFILE_ROLE

System Privileges Granted to SYS_AUTH_PERMS_ADMIN_ROLE

Each system privilege is granted with no administrative rights.

- CHANGE PASSWORD
- MANAGE ANY OBJECT PRIVILEGES
- MANAGE ROLES

Granting SYS_AUTH_PROFILE_ROLE

Allows users to enable or disable server tracing for application profiling.

Prerequisites

Administrative privilege over SYS_AUTH_PROFILE_ROLE.

Task

You can grant this role with or without administrative rights. When granted with administrative rights, you can manage (grant and revoke) the role, as well as use any of the underlying system privileges. When granted with administrative rights only, you can manage the role, but not use its underlying system privileges. Finally, when granted with no administrative rights, you can only use its underlying system privileges. By default, the SYS_AUTH_PROFILE_ROLE is granted the diagnostics system role with no administrative rights.

To grant the SYS_AUTH_PROFILE_ROLE role, execute one of the following statements:

Administrative Option	Statement
With full administrative rights	**GRANT ROLE SYS_AUTH_PROFILE_ROLE** TO *user_ID* **WITH ADMIN OPTION**
With administrative rights only	**GRANT ROLE SYS_AUTH_PROFILE_ROLE** TO *user_ID* **WITH ADMIN ONLY OPTION**
With no administrative rights	**GRANT ROLE SYS_AUTH_PROFILE_ROLE** TO *user_ID* **WITH NO ADMIN OPTION**

Example:

This example grants the SYS_AUTH_PROFILE_ROLE to `Mary`, with administrative rights.

```
GRANT ROLE SYS_AUTH_PROFILE_ROLE TO Mary WITH ADMIN OPTION
```

System Privileges Granted to SYS_AUTH_PROFILE_ROLE

The MANAGE PROFILING system privilege is granted with no administrative rights.

Granting SYS_AUTH_READFILE_ROLE

Allows users to read to a file that is resident on the server machine.

Prerequisites

Administrative privilege over SYS_AUTH_READFILE_ROLE.

Task

You can grant this role with or without administrative rights. When granted with administrative rights, you can manage (grant and revoke) the role, as well as use any of the underlying system privileges. When granted with administrative rights only, you can manage the role, but not use its underlying system privileges. Finally, when granted with no administrative rights, you can only use its underlying system privileges.
To grant the SYS_AUTH_READFILE_ROLE role, execute one of the following statements:

Administrative Option	Statement
With full administrative rights	**GRANT SYS_AUTH_READFILE_ROLE** TO *user_ID* **WITH ADMIN OPTION**
With administrative rights only	**GRANT SYS_AUTH_READFILE_ROLE** TO *user_ID* **WITH ADMIN ONLY OPTION**
With no administrative rights	**GRANT SYS_AUTH_READFILE_ROLE** TO *user_ID* **WITH NO ADMIN OPTION**

Example:

This example grants the SYS_AUTH_READFILE_ROLE to `Mary`, with no administrative rights.

```
GRANT ROLE SYS_AUTH_READFILE_ROLE TO Mary WITH NO ADMIN OPTION
```

System Privileges Granted to SYS_AUTH_READFILE_ROLE
The READ FILE system privilege is granted with no administrative rights.

Granting SYS_AUTH_READCLIENTFILE_ROLE

Allows users to read to a file that is resident on the client machine.

Prerequisites

Administrative privilege over SYS_AUTH_READCLIENTFILE_ROLE.

Task

You can grant this role with or without administrative rights. When granted with administrative rights, you can manage (grant and revoke) the role, as well as use any of the underlying system privileges. When granted with administrative rights only, you can manage the role, but not use its underlying system privileges. Finally, when granted with no administrative rights, you can only use its underlying system privileges.
To grant the SYS_AUTH_READCLIENTFILE_ROLE role, execute one of the following statements:

Administrative Option	Statement
With full administrative rights	**GRANT ROLE SYS_AUTH_READCLIENTFILE_ROLE** TO *user_ID* **WITH ADMIN OPTION**
With administrative rights only	**GRANT ROLE SYS_AUTH_READCLIENTFILE_ROLE** TO *user_ID* **WITH ADMIN ONLY OPTION**
With no administrative rights	**GRANT ROLE SYS_AUTH_READCLIENTFILE_ROLE** TO *user_ID* **WITH NO ADMIN OPTION**

Example:

This example grants the SYS_AUTH_READCLIENTFILE_ROLE to `Mary`, with only administrative rights.

```
GRANT ROLE SYS_AUTH_READCLIENTFILE_ROLE TO Mary WITH ADMIN ONLY
OPTION
```

System Privileges Granted to SYS_AUTH_READCLIENTFILE_ROLE
The READ CLIENT FILE system privilege is granted with no administrative rights.

Granting SYS_AUTH_RESOURCE_ROLE

Allows users to create new database objects, such as tables, views, indexes, or procedures.

Prerequisites

Administrative privilege over SYS_AUTH_RESOURCE_ROLE.

Task

You can grant this role with or without administrative rights. When granted with administrative rights, you can manage (grant and revoke) the role, as well as use any of the underlying system privileges. When granted with administrative rights only, you can manage the role, but not use its underlying system privileges. Finally, when granted with no administrative rights, you can only use its underlying system privileges.

Note: When migrating from SAP Sybase IQ 15.4 and earlier, the concept of inheritance of the underlying system privileges of this system role represents a change in behavior in SAP Sybase IQ 16.0 or later. For SAP Sybase IQ 15.4 and earlier behavior, use the WITH NO SYSTEM PRIVILEGE INHERITANCE clause.

The WITH ADMIN ONLY OPTION and WITH ADMIN OPTION clauses are invalid when using the WITH NO SYSTEM PRIVILEGE INHERITANCE clause. The WITH NO

ADMIN OPTION clause is valid, but not required, as it is semantically equivalent to the WITH NO SYSTEM PRIVILEGE INHERITANCE clause.

To grant the SYS_AUTH_RESOURCE_ROLE role, execute one of the following statements:

Administrative Option	Statement
With full administrative rights	**GRANT ROLE SYS_AUTH_RESOURCE_ROLE** TO *user_ID* **WITH ADMIN OPTION**
With administrative rights only	**GRANT ROLE SYS_AUTH_RESOURCE_ROLE** TO *user_ID* **WITH ADMIN ONLY OPTION**
With no administrative rights	**GRANT ROLE SYS_AUTH_RESOURCE_ROLE** TO *user_ID* **WITH NO ADMIN OPTION**
With no system privilege inheritance	**GRANT ROLE SYS_AUTH_RESOURCE_ROLE** TO *user_ID* **WITH NO SYSTEM PRIVILEGE INHERITANCE**

Example:

This example grants the SYS_AUTH_RESOURCE_ROLE to `Mary` with administrative rights and inheritance of the underlying system privileges of the role.

```
GRANT ROLE SYS_AUTH_RESOURCE_ROLE TO Mary WITH ADMIN OPTION
```

This example grants the SYS_AUTH_RESOURCE_ROLE to `Joe` with neither administrative rights nor inheritance of the roles underlying system privileges.

```
GRANT ROLE SYS_AUTH_RESOURCE_ROLE TO Joe
WITH NO SYSTEM PRIVILEGE INHERITANCE
```

System Privileges Granted to SYS_AUTH_RESOURCE_ROLE

Each system privilege is granted with no administrative rights.

- CREATE TABLE
- CREATE PROXY TABLE
- CREATE VIEW
- CREATE MATERIALIZED VIEW
- CREATE PROCEDURE
- CREATE DATATYPE

- CREATE MESSAGE
- CREATE TEXT CONFIGURATION
- CREATE ANY SEQUENCE
- CREATE ANY TRIGGER
- ALTER ANY TRIGGER
- CREATE ANY OBJECT

Granting SYS_AUTH_SPACE_ADMIN_ROLE

Allows users to manage dbspaces.

Prerequisites

Administrative privilege over SYS_AUTH_SPACE_ADMIN_ROLE.

Task

You can grant this role with or without administrative rights. When granted with administrative rights, you can manage (grant and revoke) the role, as well as use any of the underlying system privileges. When granted with administrative rights only, you can manage the role, but not use its underlying system privileges. Finally, when granted with no administrative rights, you can only use its underlying system privileges.
To grant the SYS_AUTH_SPACE_ADMIN_ROLE role, execute one of the following statements:

Administrative Option	Statement
With full administrative rights	**GRANT ROLE SYS_AUTH_SPACE_ADMIN_ROLE** TO *user_ID* **WITH ADMIN OPTION**
With administrative rights only	**GRANT ROLE SYS_AUTH_SPACE_ADMIN_ROLE** TO *user_ID* **WITH ADMIN ONLY OPTION**
With no administrative rights	**GRANT ROLE SYS_AUTH_SPACE_ADMIN_ROLE** TO *user_ID* **WITH NO ADMIN OPTION**

Example:

This example grants the SYS_AUTH_SPACE_ADMIN_ROLE to `Mary`, with no administrative rights.

```
GRANT ROLE SYS_AUTH_SPACE_ADMIN_ROLE TO Mary WITH NO ADMIN OPTION
```

System Privileges Granted to SYS_AUTH_SPACE_ADMIN_ROLE

The ACCESS SERVER LS and MANAGE ANY DBSPACE system privileges are granted with no administrative rights.

Granting SYS_AUTH_USER ADMIN_ROLE

Allows users to manage external logins, login policies, and other users.

Prerequisites

Administrative privilege over SYS_AUTH_USER ADMIN_ROLE.

Task

You can grant this role with or without administrative rights. When granted with administrative rights, you can manage (grant and revoke) the role, as well as use any of the underlying system privileges. When granted with administrative rights only, you can manage the role, but not use its underlying system privileges. Finally, when granted with no administrative rights, you can only use its underlying system privileges.

To grant the SYS_AUTH_USER ADMIN_ROLE role, execute one of the following statements:

Administrative Option	Statement
With full administrative rights	**GRANT ROLE SYS_AUTH_USER_ADMIN_ROLE** TO *user_ID* **WITH ADMIN OPTION**
With administrative rights only	**GRANT ROLE SYS_AUTH_USER_ADMIN_ROLE** TO *user_ID* **WITH ADMIN ONLY OPTION**
With no administrative rights	**GRANT ROLE SYS_AUTH_USER_ADMIN_ROLE** TO *user_ID* **WITH NO ADMIN OPTION**

Example:

This example grants the SYS_AUTH_USER_ADMIN_ROLE to `Mary`, with administrative rights.

```
GRANT ROLE SYS_AUTH_USER_ADMIN_ROLE TO Mary WITH ADMIN OPTION
```

System Privileges Granted to SYS_AUTH_USER_ADMIN_ROLE

The MANAGE ANY LOGIN POLICY and MANAGE ANY USER system privileges are granted with no administrative rights.

Granting SYS_AUTH_VALIDATE_ROLE

Allows users to validate or check tables, materialized views, indexes, or databases in the system store. The objects can be owned by any user.

Prerequisites

Administrative privilege over SYS_AUTH_VALIDATE_ROLE.

Task

You can grant this role with or without administrative rights. When granted with administrative rights, you can manage (grant and revoke) the role, as well as use any of the underlying system privileges. When granted with administrative rights only, you can manage the role, but not use its underlying system privileges. Finally, when granted with no administrative rights, you can only use its underlying system privileges.

Note: When migrating from SAP Sybase IQ 15.4 and earlier, the concept of inheritance of the underlying system privileges of this system role represents a change in behavior in SAP Sybase IQ 16.0 or later. For SAP Sybase IQ 15.4 and earlier behavior, use the WITH NO SYSTEM PRIVILEGE INHERITANCE clause.

The WITH ADMIN ONLY OPTION and WITH ADMIN OPTION clauses are invalid when using the WITH NO SYSTEM PRIVILEGE INHERITANCE clause. The WITH NO ADMIN OPTION clause is valid, but not required, as it is semantically equivalent to the WITH NO SYSTEM PRIVILEGE INHERITANCE clause.

To grant the SYS_AUTH_VALIDATE_ROLE role, execute one of the following statements:

Administrative Option	Statement
With full administrative rights	**GRANT ROLE SYS_AUTH_VALIDATE_ROLE** TO *user_ID* **WITH ADMIN OPTION**
With administrative rights only	**GRANT ROLE SYS_AUTH_VALIDATE_ROLE** TO *user_ID* **WITH ADMIN ONLY OPTION**
With no administrative rights	**GRANT ROLE SYS_AUTH_VALIDATE_ROLE** TO *user_ID* **WITH NO ADMIN OPTION**
With no system privilege inheritance	**GRANT ROLE SYS_AUTH_VALIDATE_ROLE** TO *user_ID* **WITH NO SYSTEM PRIVILEGE INHERITANCE**

Example:

This example grants the SYS_AUTH_VALIDATE_ROLE to `Mary` with administrative rights and inheritance of the underlying system privileges of the role.

```
GRANT ROLE SYS_AUTH_VALIDATE_ROLE TO Mary WITH ADMIN OPTION
```

This example grants the SYS_AUTH_VALIDATE_ROLE to `Joe` with neither administrative rights nor inheritance of the roles underlying system privileges.

```
GRANT ROLE SYS_AUTH_VALIDATE_ROLE TO Joe
WITH NO SYSTEM PRIVILEGE INHERITANCE
```

System Privileges Granted to SYS_AUTH_VALIDATE_ROLE

The VALIDATE ANY OBJECT system privilege is granted with no administrative rights.

Granting SYS_AUTH_WRITEFILE_ROLE

Allows users to write to a file that is resident on the server machine.

Prerequisites

Administrative privilege over SYS_AUTH_WRITEFILE_ROLE.

Task

You can grant this role with or without administrative rights. When granted with administrative rights, you can manage (grant and revoke) the role, as well as use any of the underlying system privileges. When granted with administrative rights only, you can manage the role, but not use its underlying system privileges. Finally, when granted with no administrative rights, you can only use its underlying system privileges.
To grant the role, execute one of the following statements:

Administrative Option	Statement
With full administrative rights	**GRANT ROLE SYS_AUTH_WRITEFILE_ROLE** TO *user_ID* **WITH ADMIN OPTION**
With administrative rights only	**GRANT ROLE SYS_AUTH_WRITEFILE_ROLE** TO *user_ID* **WITH ADMIN ONLY OPTION**
With no administrative rights	**GRANT ROLE SYS_AUTH_WRITEFILE_ROLE** TO *user_ID* **WITH NO ADMIN OPTION**

Example:

This example grants the SYS_AUTH_WRITEFILE_ROLE to `Mary`, with no administrative rights.

```
GRANT ROLE SYS_AUTH_WRITEFILE_ROLE TO Mary WITH NO ADMIN OPTION
```

System Privileges Granted to SYS_AUTH_WRITEFILE_ROLE

The WRITE FILE system privilege is granted with no administrative rights.

Granting SYS_AUTH_WRITECLIENTFILE_ROLE

Allows users to write to a file that is resident on the client machine.

Prerequisites

Administrative privilege over SYS_AUTH_WRITECLIENTFILE_ROLE.

Task

You can grant this role with or without administrative rights. When granted with administrative rights, you can manage (grant and revoke) the role, as well as use any of the underlying system privileges. When granted with administrative rights only, you can manage the role, but not use its underlying system privileges. Finally, when granted with no administrative rights, you can only use its underlying system privileges.

To grant the SYS_AUTH_WRITECLIENTFILE_ROLE role, execute one of the following statements:

Administrative Option	Statement
With full administrative rights	**GRANT ROLE SYS_AUTH_WRITECLIENTFILE_ROLE** TO *user_ID* **WITH ADMIN OPTION**
With administrative rights only	**GRANT ROLE SYS_AUTH_WRITECLIENTFILE_ROLE** TO *user_ID* **WITH ADMIN ONLY OPTION**
With no administrative rights	**GRANT ROLE SYS_AUTH_WRITECLIENTFILE_ROLE** TO *user_ID* **WITH NO ADMIN OPTION**

Example:

This example grants the SYS_AUTH_WRITECLIENTFILE_ROLE to `Mary`, with only administrative rights.

```
GRANT ROLE SYS_AUTH_WRITECLIENTFILE_ROLE TO Mary WITH ADMIN ONLY
OPTION
```

System Privileges Granted to SYS_AUTH_WRITECLIENTFILE_ROLE

The SWRITE CLIENT FILE system privilege is granted with no administrative rights.

Revoking a Compatibility Role

Revoke a compatibility role from a user or role.

Prerequisites

Administrative privilege over the compatibility role being revoked.

Task

To revoke a compatibility role, execute one of these statements:

Administrative Option	Statement
Administrative rights only	**REVOKE ADMIN OPTION FOR ROLE** *compatibility_role* **FROM** *grantee [,...]*
Membership in the role and any administrative rights	**REVOKE ROLE** *compatibility_role* **FROM** *grantee [,...]*

Migrating a Compatibility Role

Migrate all underlying system privileges of a compatibility role to a user-defined role.

Prerequisites

Administrative privilege over the role being migrated, and the MANAGE ROLES system privilege.

Task

Compatibility roles are immutable, but you can migrate them in their entirety to a new user-defined role. Once migrated, the compatibility role is automatically dropped. This process is systematically equivalent to individually granting each underlying system privilege to a user-defined role, then manually dropping the compatibility role.

During migration:

- A new user-defined role is created.
- All system privileges currently granted to the migrating role are granted to the new user-defined role.
- All users and roles currently granted to the migrating role are granted to the new user-defined role.
- Administrators of the compatibility role continue to be the administrators of the new migrated role.
- The compatibility role is dropped.

You cannot use **ALTER ROLE** to individually migrate the SYS_AUTH_SA_ROLE and SYS_AUTH_SSO_ROLE compatibility roles. These two compatibility roles are automatically migrated when SYS_AUTH_DBA_ROLE is migrated.

The name of the new role cannot already exist, or begin with the prefix SYS_ and end with the suffix _ROLE when migrating a compatibility role.

To migrate a compatibility role, execute one of these statements:

Compatibility Role	Statement
SYS_AUTH_DBA_ROLE	**ALTER ROLE** *SYS_AUTH_DBA_ROLE* **MIGRATE TO** *new_dba_role_name, new_sa_role_name, new_sso_role_name*
Any other compatibility role	**ALTER ROLE** *compatibility_sys_role_name* **MIGRATE TO** *new_role_name*

Example

The following statement migrates SYS_AUTH_DBA_ROLE to the new roles `Custom_DBA_Role`, `Custom_SA_Role`, and `Custom_SSO_Role`, respectively.

```
ALTER ROLE SYS_AUTH_DBA_ROLE
MIGRATE TO Custom_DBA_Role, Custom_SA_Role, Custom_SSO_Role
```

This statement migrates the SYS_AUTH_OPERATOR_ROLE role to the new role `Custom_Operator_role`.

```
ALTER ROLE SYS_AUTH_OPERATOR_ROLE
MIGRATE TO Custom_Operator_Role
```

In both examples, all users, underlying system privileges, and roles granted to the original roles are automatically migrated to the new roles, then SYS_AUTH_DBA_ROLE, SYS_AUTH_SA_ROLE, SYS_AUTH_SSO_ROLE and SYS_AUTH_OPERATOR_ROLE are dropped.

Dropping a Compatibility Role

All compatibility roles, with the exception of SYS_AUTH_SA_ROLE and SYS_AUTH_SSO_ROLE can be dropped. SYS_AUTH_SA_ROLE and SYS_AUTH_SSO_ROLE are automatically dropped when SYS_AUTH_DBA_ROLE is dropped.

Prerequisites

- Administrative privilege over the role being dropped.
- Each of the system privileges granted to a compatibility role is granted to at least one user-defined role.

Task

The **WITH REVOKE** clause is only required when dropping a compatibility role that is currently granted to users or roles. You can re-create any dropped compatibility role. See *Re-creating Compatibility Roles*.

To delete a compatibility role, execute one of these statements:

Drop Condition	Statement
Compatibility role not currently granted to any user or role.	**DROP ROLE** *role_name*[*]
Compatibility role currently granted to users or roles.	**DROP ROLE** *role_name*[*] **WITH REVOKE**

[*] *role_name* cannot be SYS_AUTH_SA_ROLE or SYS_AUTH_SSO_ROLE.

See also

- *Re-creating a Compatibility Role* on page 55

Re-creating a Compatibility Role

Re-creates a dropped compatibility role.

Prerequisites

- MANAGE ROLES system privilege.
- Administrative privileges on all of the system privileges granted to the compatibility role being re-created.

Task

Re-creating SYS_AUTH_DBA_ROLE is semantically equivalent to re-creating both the SYS_AUTH_SA_ROLE and SYS_AUTH_SSO_ROLE roles; you cannot re-create these two roles separately.

When you re-create any compatibility role other than SYS_AUTH_DBA_ROLE, administrative privileges on the re-created compatibility role are automatically granted to SYS_AUTH_DBA_ROLE, as long as it still exists.

When you re-create any compatibility role other than SYS_AUTH_DBA_ROLE, or SYS_AUTH_PERMS_ADMIN_ROLE, administrative privileges on the re-created compatibility role are automatically granted to SYS_AUTH_PERMS_ADMIN_ROLE, as long as it still exists.

To re-create a compatibility role, execute:

CREATE ROLE *compatibility_role_name* [**WITH ADMIN [ONLY]** *userid [, ...]*]

Implications of Migrating Compatibility Roles

Some system roles are indirectly granted the system privileges necessary to execute privileged tasks through membership in compatibility roles.

You cannot revoke the underlying system privileges of a compatibility role; you must first migrate it to a user-defined role. Only then can you revoke individual underlying system privileges from the new role and grant them to other user-defined roles per the organization's security requirements. This enforces separation of duties.

You can migrate compatibility roles automatically or manually. The method of migration can impact the ability of a system role to continue performing authorized tasks.

Regardless of the migration method used, once a compatibility role is dropped, if you revoke a system privilege from the new user-defined role and grant it to another user-defined role, you must do one of the following to ensure that system roles retain all the system privileges required to execute applicable privileged tasks:

- Grant each system privilege revoked from the migrated user-defined role directly to the system roles; or
- Grant membership in the user-defined role to which the system privileges are granted to the system roles.

The system roles that are members of compatibility roles, and might potentially be impacted by migration include:

System Role	Compatibility Role
dbo	SYS_AUTH_DBA_ROLE SYS_AUTH_RESOURCE_ROLE
SYS_RUN_REPLICATION_ROLE	SYS_AUTH_DBA_ROLE

Automatic Migration

The **ALTER ROLE** statement creates a new user-defined role, automatically grants all underlying system privileges of the compatibility role to the new user-defined role, makes each member of the compatibility role a member of the new user-defined role, then drops the compatibility role.

Automatic migration assumes that the destination user-defined role does not already exist and that all system privileges are migrated to the same new user-defined role.

Manual Migration

Use the **CREATE ROLE** statement to create a new user-defined role. Use the **GRANT** statement to grant each underlying system privilege to one or more users or roles. Use the **DROP** statement to drop the compatibility role once all underlying system privileges are granted to at least one other user or role.

Members of the migrated compatibility role are not automatically granted membership in the new user-defined role. As a result, members of some system roles may no longer be able to perform the expected privileged tasks once the compatibility role is dropped. You must grant membership in the new user-defined role to the affected system roles or directly grant the required system privileges to affected members.

Backward Compatibility in SAP Sybase IQ 16.0

Grant and revoke syntax for role-based security differs significantly from authority-based security. However, SAP Sybase IQ 16.0 is fully backward compatible with authority-based syntax.

SAP Sybase IQ 16.0 provides well-documented mappings and stored procedures to assist in transition. All stored procedures, functions, and queries created in pre-16.0 databases will continue to run after upgrading.

Stored Procedure to Map Authorities to System Roles

The **sp_auth_sys_role_info** stored procedure generates a report, which maps each authority to a corresponding system role name.

A separate row is generated for each authority. No privilege is required to execute the procedure.

Connecting to SAP Sybase IQ 15.x Databases with SAP Sybase IQ 16.0

Role-based syntax is not supported in SAP Sybase IQ 15.x databases.

When using SAP Sybase IQ 16.0 to connect to a 15.x database, only authority-based syntax is valid. Using role-based syntax returns errors. For example, GRANT ROLE returns an error message; GRANT MEMBERSHIP IN GROUP does not.

Beyond this limitation, there should be no change in functionality and no noticeable change in performance using SAP Sybase IQ 16.0 with a 15.x database.

SAP Sybase IQ 12.7 Database Migration

Updating a 12.7 catalog to 16.0 SPx requires a database file migration, not a simple database upgrade. Migration tools bundled with SAP Sybase IQ 16.0 SPx can recreate the legacy database schema and database options.

Preparing to Migrate

Perform these tasks before you migrate your database.

1. Upgrade to SAP Sybase IQ 12.7 ESD #5. All migration paths assume that you are migrating from SAP Sybase IQ 12.7 ESD #5 as a minimum.
2. Regenerate any sort-key values. SAP Sybase IQ 12.7 **SORTKEY** function uses a different sort-key value than SAP Sybase IQ 15 and later.

3. Review the collation. SAP Sybase IQ no longer supports custom collations. Custom collations are preserved in database rebuilds only if you rebuild the database in a single step. Use a collation included with SAP Sybase IQ 15.0 or later.
4. Back up your current installation and save copies of any changes you made to default login and post login scripts. Create your backups on removable media, like tape, DVD, or CD. Make sure the backups are readable.
5. Review and understand the database migration utilities. Use **iqunload** to re-create the schema for your database or migrate your 12.7 database. **iqlsunload** utility is available to move 12.7 local stores for 12.7 multiplex servers.
6. Use **DROP TABLE** statements to drop all global temporary tables before you run the **iqunload** utility. You can recreate the global temporary tables after migration.
7. Drop all servers of type `asejdbc` before you run the **iqunload** utility.

   The `asejdbc` server class is deprecated. Servers of type `asejdbc` must be dropped prior to running iqunload. 16.0 12.7 databases with remote server definitions based on the `asejdbc` driver will not have these definitions migrated to by the **iqunload** tool and will not give an error at the beginning of an unload saying that you need to drop any asejdbc servers (if there are any present).
8. Resolve potential migration errors. SAP Sybase IQ no longer supports some 12.7 features and objects. Update these objects before you migrate the database.
9. Use **sp_iqcheckdb** to verify that your 12.7 database is clean and error free.

   For information on **sp_iqcheckdb** output, see *Administration: Backup, Restore, and Data Recovery > System Recovery and Database Repair*.

Migration Utilities

Utilities and support tools bundled with SAP Sybase IQ.

iqunload Utility

iqunload is a command line utility for SAP Sybase IQ 12.6 and 12.7 database migration. **iqunload** re-creates the legacy catalog on the new database catalog in the current installation.

iqunload does not change SAP Sybase IQ data and temp dbspaces. The utility preserves all legacy database options and applies them to the new database. SAP Sybase IQ ignores any legacy options that no longer apply to the current version of the software.

Note: If you are reloading a 12.6 or 12.7 database, remove any square brackets or back quotes in identifiers; otherwise, the reload fails.

Syntax

```
iqunload [ options ]  directory [ @data ]
```

```
data:[ environment variable | file ]
```

Parameters

iqunload takes one or more parameters.

- **-ap <*size*> –** (optional) Sets the page size for the new catalog store.
- **-au –** required for migration mode. Migrates the database.

  Specify an **-au** argument to start **iqunload** in migration mode. or **-n** argument, but not both.
- **-c "*keyword* = *value*, ..." –** (required) Supply database connection parameters. You must specify a DBF parameter to specify the name of the database file for migration. The file path is either absolute or relative to the server startup directory.
- **-dc –** (optional) Recalculate computed columns in the database.
- **-ms_filename –** optional for simplex migration; required for multiplex migration. Use **-ms_filename** to specify a file name for the new empty `IQ_SYSTEM_MAIN` store created during the migration.

  If not specified, the default new main store is a file system file called `new_main_store.iq`
- **-ms_reserve –** (optional) Specifies the size of the new `IQ_SYSTEM_MAIN` reserve, in MB. If unspecified, defaults to zero.
- **-ms_size –** (optional) specifies the size of the new `IQ_SYSTEM_MAIN` store, in MB, based on the database size. The minimum, assuming a default page size, is 200MB. If you specifiy an **–ms_size** value smaller than the computed value, SAP Sybase IQ uses the computed value; otherwise the specified value is used.
- **-n –** required for schema unload only. Unloads the schema definition only. The **-n** parameter requires 12.7 ESD #5 or later. Specify **-au** or **-n** argument but not both.
- **-new_startline –** (optional) specify startup switches for the new server that is the migration target. For a complete list of server startup switches, see the *Utility Guide*.
- **-o *`file name`* –** (optional) logs output messages to *file name*.
- **-q –** (optional) suppresses messages and windows.
- **-r *file name* –** (optional) specifies the file name.
- **-t *list* –** (optional) outputs listed tables only. Can specify `OwnerName.TableName` or `TableName` alone.
- **-v –** (optional) returns verbose messages.
- **-y –** (optional) replaces existing reload schema SQL script with new output without confirmation.

Examples

- **Example 1 –** migrates a simplex database to a current server.

```
iqunload –au –c
“UID=DBA;PWD=SQL;DBF=/mydevice/test_dir/test2.db”
```

- **Example 2 –** unloads a legacy database schema and renames the generated SQL script to `test2_reload.sql`:

  ```
  iqunload -n -c “UID=DBA;PWD=SQL;DBF=D:\\test_dir\\test2\
  \test2.db;
  ```

- **Example 3 –** migrates database `test3.db`. The **START** connection parameter specifies switches for starting the database being reloaded. The **-new_startline** parameter specifies switches for starting the `utility_db` to create the new catalog store.

  ```
  iqunload -au -c
  "UID=DBA;PWD=SQL;DBF=test3.db;START=-ch 128M -iqmc 50" -
  new_startline "-ch 256M -iqtc 400"
  ```

  Do not include **-iqnotemp** *XYZ* in the new start line or migration fails. As part of the migration process, temp files are added to `IQ_SYSTEM_TEMP`. If you start the server with the **-iqnotemp** option, **iqunload** cannot add these temp files

- **Example 4 –** migrates the legacy database, `asiqdemo.db`, using a raw device for the `IQ_SYSTEM_MAIN` store:

  ```
  iqunload -au -c
  "UID=DBA;PWD=SQL;DBF=asiqdemo.db" -ms_filename "/dev/rdsk/
  c4t0d0s3"
  ```

Usage

iqunload has two working modes: schema unload and migration.

Schema Unload Mode

iqunload requires an **-n** argument to start in schema unload mode. Schema unload mode unloads a 12.6 ESD #11 or 12.7 ESD #5 database schema, and generates a script (`reload.sql`) that can re-create the schema for a database in a current version of the software. A **-c** argument is required for connection parameters:

```
iqunload -n -c "UID=DBA;PWD=SQL;ENG=my_engine;DBN=my_dbname"
```

Schema unload mode re-creates the schema, but does not migrate data. To migrate data, extract the legacy data and load the new database.

Migration Mode

iqunload requires an **-au** argument to start in migration mode. **iqunload** migration mode interfaces with the 12.7 support engine (**iqunlspt**) and the current database server (**iqsrv16**):

- Start the legacy database and generate the schema
- Start the current SAP Sybase IQ server
- Create a new database and apply the legacy schema

General Usage

- Insufficient cache memory causes migration errors. **iqunload** uses default values for various cache sizes (catalog cache, main buffer cache, temp cache). If the legacy database requires higher cache values, use the **–ch** and **-cl** options as part of the **START** connection parameter to increase the cache size.
- During database migration, the server creates a message file (`*.iqmsg.R`) as it reloads the generated schema. This file is normally deleted as part of a cleanup operation for successful migrations. If the migration fails during the reload stage, cleanup does not occur, and `*.iqmsg.R` remains in the `unload` directory. `*.iqmsg.R` may contain information that can help solve your migration problems.
- **iqunload** writes some temporary files to the $IQTMP16 directory. If you set the $IQTMP16 environment variable, set it to a valid directory name.
- Users with wide tables (large numbers of column/null values) should not decrease the catalog store page size for database migration.
- If the legacy database contains invalid views, SAP Sybase IQ completes the migration but issues warnings. A warning may occur, for example, if the tables involved in a view are dropped.
- If the legacy database is encrypted, use the **DBKEY** connection parameter to provide the encryption key. The migrated database uses the same encryption key.

Permissions

DBA

See also

- *iqlsunload Utility* on page 61
- *Support Processes* on page 64

iqlsunload Utility

In current multiplex configurations, multiple nodes can write to the main store, which eliminates the need for local stores. **iqlsunload** is a command line utility that you can use to unload a 12.7 local store. **iqlsunload** is used only in 12.7 ESD #5 multiplex migrations.

iqlsunload is bundled with all versions of SAP Sybase IQ starting with 12.7 ESD #5.

Syntax

```
iqlsunload [ options ]  directory [ @data ]
data:[ environment variable | file ]
```

Parameters

- **directory –** (required) identifies the directory where **iqlsunload** unloads the data files. Create this directory before you run **iqlsunload**, or point to an existing directory. This directory must be relative to the database on the database server.

- **-al** **–** (optional) unloads IQ local store schema and data.
- **-c *“keyword=value;...”* –** (optional) supplies database connection parameters.
- **-h –** (optional) prints out the syntax (help) for the utility.
- **-o *filename* –** (optional) logs output messages, including errors, to *filename*.
- **-q –** (optional) suppresses messages and windows.
- **-r *directory* –** (optional) specifies the directory where SQL scripts are generated. The default reload command file is `reload.sql` in the current directory. The directory is relative to the current directory of the client application, not the server.
- **-t *list* –** (optional) outputs listed tables only. Can specify `OwnerName.TableName` or `TableName` alone. Cannot be specified with **al** argument.
- **-v –** (optional) outputs verbose messages.
- **-y –** (optional) replaces existing reload schema SQL script without confirmation.

Examples

- **Example 1 –** unload local stores from a database called `mpxtest2`, extracting any table data to the directory `/mydevice/test_dir/unload_dir` :

```
iqlsunload -o iqunload_624.out -al
-c “UID=DBA;PWD=SQL;ENG=myserver_mpxtest02”
/mydevice/test_dir/unload_dir
```

Usage

General Notes

- Run **iqlsunload** from the `$IQDIR16/lsunload` directory to pick up updated libraries before resolving any IQ 12.7 libraries.

Unloaded Objects

Running **iqlsunload** with an **-al** argument unloads these persistent objects:

- Base tables
- Global temporary tables
- Indexes
- Domains (user-defined data types)
- Constraints (column check constraint, table constraint, primary key, foreign key, unique, default, IQ unique, not null)
- Views
- Stored procedures and functions
- Messages
- Remote servers and external logins
- Events

Empty User Names

SAP Sybase IQ no longer allow users with empty user names. You cannot drop or migrate users with empty user names the 12.6 or 12.7 server. The schema reload operation warns that an empty user name has been encountered and that the user will not be re-created. The reload operation ignores such users and any associated objects.

Unloading Tables

iqlsunload ignores any system tables or nonexistent tables:

- If you extract table schema and data only, the legacy database collation must match the collation of the current database collation.
- If you do not qualify table names with owner names, **iqlsunload** extracts table data from all tables with that table name.

Output Files

iqlsunload generates these output files:

Script Name	Description
`reload_schema.sql`	Recreates schema for unloaded objects (either objects from local store or tables selected by the user.) This script is executed against a node that writes to the multiplex. This node can be either the existing writer node, or a writer or coordinator for the multiplex after migration, depending on where you plan to recreate the schema.
`extract_data.sql`	Extracts table data for the unloaded tables from the local store. Execute this script in Interactive SQL while connected to the query node from which it was generated. When this script executes, it generates the data files into the directory data.
`reload_data.sql`	Loads extracted table data. This script is executed on the node where you ran `reload_schema.sql`, and reloads the data extracted from the `extract_data.sql` file.

Permissions

DBA

See also

- *iqunload Utility* on page 58
- *Support Processes* on page 64

Support Processes

Running **iqunload** in migration mode (**-au**) starts **iqunlspt** and **iqsrv16**.

iqunlspt

iqunlspt is a self-contained subset of the SAP Sybase IQ 12.7 (ESD #5) database engine. It runs as a background process and provides support for legacy database unloads. **iqunlspt** starts on your legacy database with these options as defaults:

```
iqunlspt -iqnotemp 100 -iqro 1 -c 48MB -gc 20 -gd
all -gk all -gl all -gm 1 -gu all -ti 4400 -x shmem .
```

If your database requires special switches or memory setting, **iqunlspt** accepts additional startup arguments. See the *Utility Guide*.

Default cache settings are sufficient for most migrations. At migration, data queries execute against the system catalogs, not IQ data, so the **iqunlspt** engine needs lower cache levels than complex queries or multiple concurrent users. The amount of time required to start the legacy database is the same as to start **iqunlspt**. This time is included in the **iqunload** startup time.

iqsrv16

iqunload starts **iqsrv16** with these options:

```
iqsrv16 -gp 4096 -c 40p -gc 20 -gd all -gk all -gl all
-gm 1 -gu all -qi -qs -ti 4400
```

iqsrv16 also includes the **-n** parameter followed by a special randomly generated server name. The **-c 40p** setting provides a larger cache for the catalog store, allowing the server engine to execute many schema DDL statements. Both server start commands use the default values for **-iqmc** and **-iqtc**. If the legacy server requires larger startup values, use the **-c** switch to increase the server cache memory.

See also

- *iqunload Utility* on page 58
- *iqlsunload Utility* on page 61

Migration Issues

SAP Sybase IQ no longer supports some legacy features. Run **iqunload** in schema unload mode to generate a script `(reload.sql)` that contains the entire database schema. Compare the contents of this file to find unsupported syntax and metadata.

Unsupported Objects

Check the schema for objects SAP Sybase IQ no longer supports.

Table 3. Unsupported metadata

Object	Details	Action
Invalid database, table, or user names	Table names cannot contain double quote characters. User names and database names cannot contain double quote characters, single quote, or semicolon characters. User names and database names cannot start or end with a space.	Change the object name.
Reserved logical server names	A logical server cannot be named ALL, AUTO, COORDINATOR, DEFAULT, OPEN, or SERVER.	Drop the logical server before upgrading.
Join indexes	Join indexes are no longer supported.**iqunload** does not run if the database to be migrated contains join indexes.	Drop all join indexes before migrating data.
LD indexes	LD indexes are no longer supported.	Drop all LD indexes before migrating data.
Database with BLANK PADDING OFF	**iqunload** searches BLANK PADDING OFF databases for any indexes that would become invalid after migration. **iqunload** fails and lists indexes and constraints that must be dropped and in which order.	Drop these indexes and constraints before the schema reloads and recreate the indexes and constraints after schema reload has been completed.
Unenforced constraints	**iqunload** fails and lists unenforced constraints that must be dropped.	Drop unenforced constraints before proceeding with migration.
Old 1-byte FP or old 2-byte FP indexes	Databases created with SAP Sybase IQ 12.4.2 or earlier may have these indexes. Because these indexes were automatically created by SAP Sybase IQ, you cannot drop and recreate them; you must rebuild them.	Allow **iqunload** to check for these and list them. Rebuild these indexes using **sp_iqrebuildindex** before migration.The rebuilt indexes are upgraded.

See also

- *Syntax Changes* on page 66
- *Schema Size* on page 67
- *Output Logs* on page 68
- *Data Storage Changes* on page 69
- *Post-Migration Files* on page 70

Syntax Changes

Review the reload script `(reload.sql)` for legacy syntax that can cause **iqunload** to fail.

Table 4. Troubleshooting Syntax Changes

Problem	Solution
A **DECLARE LOCAL TEMPORARY TABLE** statement in a procedure or trigger causes a syntax error if the table name is prefixed with an owner name.	Remove the owner name.
If a **CREATE TRIGGER** statement does not include an owner name for the table on which the trigger is defined, and the table must be qualified with an owner when referenced by the user executing the reload.sql file, the statement fails with a `‘table-name’ not found` error.	Prefix the table name with the owner name.

If an object name (such as a table, column, variable, or parameter name) corresponds to a reserved word introduced in a later version of SAP Sybase IQ, the reload fails. (For reserved words, see *Reference: Building Blocks, Tables, and Procedures*. For example:
`CREATE PROCEDURE p( )`
`BEGIN`
`DECLARE NCHAR INT;`
`SET NCHAR = 1;`
`END`
	Change all references to the reserved word to use a different name. For variable names, prefixing the name with @ is a common convention that prevents naming conflicts.

Views that use Transact-SQL® outer joins (by specifying *= or =*) may not be created properly when they are reloaded.	Add the following line to the reload script: `SET TEMPORARY OPTION tsql_outer_joins='on'` Also set this option for your database. Rewrite any views or stored procedures that use Transact–SQL outer joins.
Stored procedures that use Transact–SQL outer joins may not work correctly.	Rewrite views and stored procedures.
Functions that have `OUT` or `INOUT` parameters cannot be reloaded.	`OUT` and `INOUT` parameters are no longer supported. Drop these functions before reloading.

See also

- *Unsupported Objects* on page 65
- *Schema Size* on page 67
- *Output Logs* on page 68

- *Data Storage Changes* on page 69
- *Post-Migration Files* on page 70

Schema Size

Increase the cache memory to migrate large and extremely large schemas.

Large Schemas

Default cache settings for large schemas may be too small and can exhaust dynamic memory in the **iqsrv16** server. Use the **-c** switch to increase the server cache memory and **-new_startline** to pass the switch to the server.

A diagnostic example includes these switches:

```
-ca 1
-c 1000m
-o /iq15outputdir/iq16console.out
```

- **-ca 1** – enables dynamic catalog cache sizing, and logs memory cache statistics to the console.
- **-c 1000m** – sets the initial catalog cache at 1GB.
- **-o /outputdir/iq16console.out** – specifies the log file for console output.

Use a text editor to xamine the `.out` file log entries. Watch how the catalog store adjusts the cache and determines if the setting is appropriate.

Note: The value shown for **-c** is in bytes. Set switches appropriately for your system. To specify megabytes, use the m suffix, as shown.

Extremely Large Schemas

Running **iqunload** in schema unload mode (**iqunload -n**) generates a single script (`reload.sql`) that includes the entire legacy schema. In some cases, you may need to break a very large `reload.sql` file into pieces that can be executed sequentially. This also helps the server manage the cache.

If **iqunload** fails in migration mode (**iqunload -au**) because dynamic memory is exhausted, set the cache settings as high as your hardware and operating system limitations allow. If the failure continues, contact SAP Sybase for assistance.

See also

- *Unsupported Objects* on page 65
- *Syntax Changes* on page 66
- *Output Logs* on page 68
- *Data Storage Changes* on page 69
- *Post-Migration Files* on page 70

Output Logs

Check the output logs to isolate migration problems.

SAP Sybase IQ 16.0 Engine Logs

Use the **-new_startline** " **-z -zr all**" argument to start **iqsrv16** with extra logging:

```
iqunload -au -c "UID=DBA;PWD=SQL;DBF=/iq-15/
unload/127/db/iq127db.db" -new_startline "-z -zr all"
-o iq15db.out
```

SAP Sybase IQ 12.7 Engine Logs

Use the **START** = **-z -zr all** argument to start the 12.7 engine with extra logging:

```
iqunload -v -au -c "UID=DBA;PWD=SQL;DBF=/iq-15/
unload/127/db/iq127db.db;START=-z -zr all
-o iq127db.out"
```

See the *Utility Guide* for details about the **-z** and **-zr all** parameters.

Server Not Found

A message similar to this indicates that **iqunload** started the database but could not connect to the server:

```
SQL error:Database server not found
```

Check to see if **iqunlspt** is running and stop the process before retrying **iqunload**.

For example, here is **top** output:

```
load averages:  1.45,  1.19,  0.80; up 3+16:22:31
10:2
172 processes: 168 sleeping, 2 zombie, 2 on cpu
CPU states: 79.1% idle, 18.9% user,  1.9% kernel,  0.0%
iowait,  0.0% swap
Memory: 16G phys mem, 13G free mem, 16G swap, 16G free
swap

PID USERNAME LWP PRI NICE  SIZE   RES STATE    TIME    CPU COMMAND
21223 ybrown 1  59    0 2908K 1844K cpu      0:00  0.12% top
21172 ybrown 476  59    0  319M  264M sleep    0:01  0.05% iqunlspt
24890 ybrown 14  29   10   79M   43M sleep   0:49  0.03% java
20103 ybrown 1  59    0 7916K 2080K sleep    0:00  0.00% sshd
```

To stop the process, enter the command **kill -9** and supply the process ID:

```
kill -9 21172
```

Trying to run **iqunload** without killing an orphaned **iqunlspt** process, may generate this error:

```
SQL error: Unable to start specified database: autostarting database
failed.
```

Obsolete Stored Procedures

Migration replaces 12.7 login procedures with new login management functions.

sp_login_environment replaces the 12.7 default login procedure **DBA.sp_iq_process_login** and **dbo.sa_post_login_procedure** replaces the 12.7 default post-login procedure **DBA.sp_iq_process_post_login**. **iqunload** generally replaces obsolete options with new defaults, but if the 12.7 option is set on a specific user instead of PUBLIC (the default), the log file may report errors:

```
E. 10/31 16:53:40. Login procedure
'DBA.sp_iq_process_login' caused SQLSTATE '52W09'
E. 10/31 16:53:40. Procedure 'sp_iq_process_login' not
found
```

See also

- *Unsupported Objects* on page 65
- *Syntax Changes* on page 66
- *Schema Size* on page 67
- *Data Storage Changes* on page 69
- *Post-Migration Files* on page 70

Data Storage Changes

SAP Sybase IQ 16.0 migration creates a new catalog store and changes some legacy options.

Dbspaces

In current versions of SAP Sybase IQ, all user data should reside in a user dbspace comprised of one or more files. Migration converts main dbspaces into files under one user dbspace: `IQ_MAIN`, for the SAP Sybase IQ main store, and temporary dbspaces into files under a single temporary dbspace, `IQ_SYSTEM_TEMP`, for a single SAP Sybase IQ temporary store. Existing catalog store dbspaces remain as dbspaces with a single file. All of the old main dbspaces become files in the new `iq_main` user main dbspace. Migration sets the `PUBLIC.default_dbspace` option to the value `iq_main`.

Logical names for files created from converted dbspaces are the dbspace name followed by an underscore and the file ID. For example, a main dbspace with file ID 16384 becomes `IQ_SYSTEM_MAIN_16384`.

Main Store

Migration creates a new system file for the `IQ_SYSTEM_MAIN` dbspace that contains no tables. By default, the name of this file is `new_main_store.iq`, but you can use the **ms_filename** argument to specify a different file name. The **iqunload** utility computes the size of the new `IQ_SYSTEM_MAIN` based on the size of your existing database.

If you accept the default settings for **iqunload -au -c**, the new store marked as `MAIN` has `DBSpaceName = IQ_SYSTEM_MAIN`, `DBFileName = IQ_SYSTEM_MAIN` and `path = new_main_store.iq`. For multiplex migration, the location of the new main store must be visible to all nodes on the multiplex, and you must use the **-ms_filename** argument to specify the path instead of the default value of `new_main_store.iq`.

When you migrate a database, specify the file to use for the new IQ_SYSTEM_MAIN dbspace, its name, whether or not to use a raw device, and the size of the main store and its reserve.

Migrating IQ_SYSTEM_MAIN

Specify the IQ_SYSTEM_MAIN size in the database migration command. The **-ms_size** parameter requires a value in MB, not GB. Omit **-ms_size** and **-ms_reserve** to specify a raw device. For a raw device, you must specify an unused raw partition.

This statement creates an IQ_SYSTEM_MAIN on a raw device:

```
iqunload -au -ms_filename /dev/rdsk/c1t0d1 -c
"UID=DBA;PWD=SQL;DBF=latest.db"
```

See also

- *Unsupported Objects* on page 65
- *Syntax Changes* on page 66
- *Schema Size* on page 67
- *Output Logs* on page 68
- *Post-Migration Files* on page 70

Post-Migration Files

iqunload generates a set of files derived from the legacy database. .

Table 5. Pre-Migration and Post-Migration Files

Pre-Migration	Post-Migration Files	Description
asiqdemo.db	asiqdemo.db.before_schema_reload	The 12.7 catalog database. This file is copied at the OS level upon successful migration; it is not a result of the SQL **backup** command.
asiqdemo.log	asiqdemo.log	The database log file is regenerated when the migrated database is used with the 16.0 server.
asiqdemo.iq	asiqdemo.iq	The old SAP Sybase IQ 12.7 IQ_SYSTEM_MAIN dbspace. This file and all other user dbspaces are unaffected by the migration process. This dbspace is added as a file to a user main dbspace.
asiqdemo.iqtmp	asiqdemo.iqtmp	The IQ_SYSTEM_TEMP dbspace. No operations are performed on this dbspace during migration. This file becomes the IQ 16.0 database temporary store.
asiqdemo.iqmsg	asiqdemo.iqmsg.before_schema_reload	The IQ 12.7 message file. This file is copied at the OS level upon successful migration.

Pre-Migration	Post-Migration Files	Description
	`asiqdemo.db`	The new 16.0 migrated catalog database.
	`new_main_store.iq`	The new `IQ_SYSTEM_MAIN` dbspace for the migrated database.

See also

- *Unsupported Objects* on page 65
- *Syntax Changes* on page 66
- *Schema Size* on page 67
- *Output Logs* on page 68
- *Data Storage Changes* on page 69

Unloading Legacy Schemas

To unload legacy schema, run **iqunload** in schema unload mode (**iqunload -n**) on the same machine as the legacy schema.

1. Copy these files from *`$IQDIR16`*`/unload` to *`$ASDIR`*`/scripts`:
    - `unloadold.sql`
    - `unload.sql`
    - `optdeflt.sql`
    - `opttemp.sql`
2. Start the legacy server.
3. Run **iqunload** in schema unload mode (**iqunload -n**).

    Include the appropriate connection parameters and other startup options. Schema unload mode creates a SQL script (`reload.sql`) in the current directory that contains the legacy database schema. `reload.sql` does not contain any checkpoints.For very large schemas, edit `reload.sql`, to add a few checkpoints. If you do not include extra checkpoints, IQ generates additional metadata objects that requires extra (**-iqmc**) main cache memory.

    `reload.sql` also contains a **CREATE DATABASE** template command that is commented out.
4. Create a new 16.0 database.

    Set the IQ SIZE and TEMPORARY SIZE clauses to create an IQ_SYSTEM_MAIN of 10GB and IQ_SYSTEM_TEMP of 5GB. For example:

```
CREATE DATABASE 'test.db'
IQ PATH 'test.iq'
IQ SIZE 10240
```

```
TEMPORARY PATH 'test.iqtmp'
TEMPORARY SIZE 5120
```

5. Start and connect to the new database.
6. Run the `reload.sql` against the new database.

   Execution time roughly approximates the actual time to allow for database migration, excluding validation checks. Correct any errors. Perform this process iteratively until you can cleanly load the legacy schema.

Migrating Legacy Databases

Run **iqunload** in database migration mode (**iqunload -au**) to migrate a legacy database.

Simplex Migration

Migrate a 12.7 database simplex database to 16.0.

1. *Migrating the Legacy Database*

   Make sure that the database file is not in use, and run the iqunload utility with the -au (migrate database) and -c (connection parameters).
2. *Verifying the Migrated Database*

   To verify simplex migration, start the migrated database in read-only mode and perform post-migration tasks.

Migrating the Legacy Database

Make sure that the database file is not in use, and run the **iqunload** utility with the **-au** (migrate database) and **-c** (connection parameters).

This command migrates the simplex database `mytest` and saves output in `unload.out` in the current directory:

```
iqunload -au -c "uid=DBA;pwd=SQL;dbf=mytest" -o unload.out
```

The database and the **iqunload** utility must be on the same machine to migrate the database, or **iqunload** returns an error. **-o** is an optional switch that sends a copy of the console output to the specified log file, here named `unload.out`.

Because the example specified `DBF=mytest.db`, the **iqunload** utility attempts to connect to this database in the current directory. You can also specify the full path to the database, as shown here:

```
iqunload -au -c
"dbf=/ybrown/iq-15/unload/iq127db.db;uid=DBA;pwd=SQL"
Output:
    Sybase IQ Unload Utility Version 15.0.0.5533
    Connecting and initializing
    Unloading user and group definitions
    Unloading table definitions
    Unloading index definitions
    Unloading functions
    Unloading view definitions
    Unloading procedures
    Unloading triggers
    Unloading SQL Remote definitions
    Creating new database
    Creating indexes for (1/14)
        "DBA"."sales_order"
```

```
    Creating indexes for (2/14)
        "DBA"."sales_order_items"
    Creating indexes for (3/14) "DBA"."contact"
    Creating indexes for (4/14) "DBA"."customer"
    Creating indexes for (5/14) "DBA"."fin_code"
    Creating indexes for (6/14) "DBA"."fin_data"
    Creating indexes for (7/14) "DBA"."product"
    Creating indexes for (8/14) "DBA"."department"
    Creating indexes for (9/14) "DBA"."employee"
    Creating indexes for (10/14)"DBA"."alt_sales_order"
    Creating indexes for (11/14) "DBA"."alt_sales_order_items"
    Creating indexes for (12/14) "DBA"."iq_dummy"
    Creating indexes for (13/14) "DBA"."emp1"
    Creating indexes for (14/14) "DBA"."sale"
```

```
Successfully backed up file "/ybrown/iq-15/
unload/127/db/iq127db.db" by renaming it to "/ybrown/iq-15/unload/
127/db/iq127db.db.before_schema_reload".
Successfully backed up file "/ybrown/iq-15/unload/127/db/
iq127db.iqmsg"
by renaming it to "/ybrown/iq-15/unload/127/
db/iq127db.iqmsg.before_schema_reload"
Successfully reloaded schema of database "/ybrown/iq-15/unload/127/
db/iq127db.db".
```

Perform post migration tasks. Make sure that the migration completed correctly. Back up your new databases.

Verifying the Migrated Database

To verify simplex migration, start the migrated database in read-only mode and perform post-migration tasks.

1. Start the 16.0 database in read-only mode:

   ```
   start_iq -iqro 1
   ```

   When starting the coordinator in 16.0, use the same port as the 12.7 writer server.
2. Issue a **CHECKPOINT** command.
3. Run the 16.0 version of **sp_iqcheckdb** in verify mode:

   ```
   sp_iqcheckdb ('verify database')
   ```

   If you run the procedure from Interactive SQL, redirect output to a file by entering:

   ```
   dbisql -c "..." "sp_iqcheckdb ('verify database')" >& filename
   ```

   where “...” represents startup parameters for your database.

4. Issue a **COMMIT** statement.
5. Check **sp_iqcheckdb** results for errors.

   If there is an error, you can revert to the previous database as long as you do not restart the database in write mode. To revert to the 12.7 catalog , copy all the `.before_schema_reload` files to the same file without the `.before_schema_load file` extension.
6. After you perform the read-only checks, stop the database server and restart in write mode.

Note: For information on interpreting **sp_iqcheckdb** results and corrective action, see *Administration: Backup, Restore, and Data Recovery > System Recovery and Database Repair*.

Multiplex Migration

Migrate multiplex databases, performing all steps in sequence.

1. *Synchronizing the Multiplex Nodes*

   Check the SQL Remote and multiplex server log files for synchronization problems.
2. *Migrate Local Stores*

   To move the 12.7 local stores before migration, use iqlsunload.
3. *Start the Multiplex Write Server*

   To clean the internal state information, start the write server in single-node mode.
4. *Multiplex Migration Parameters*

   To migrate the multiplex, run iqunload with the appropriate parameters.
5. *Verifying the Migrated Multiplex Database*

   Verify the migrated database in read-only mode and correct any errors.
6. *Starting the Coordinator*

   Starting the multiplex coordinator in single-node mode (-iqmpx_sn) and read-only (-iqro) performs some initial database checks. For coordinators, the server must reset an identity cookie before you can use the multiplex.
7. *Manually Synchronize the Secondary Nodes*

   To start the secondary nodes, install SAP Sybase IQ 16.0, then synchronize from the coordinator node. When you migrate a query node, it becomes a reader node.
8. *Start the Secondary Nodes*

   To start the secondary nodes, all nodes of the multiplex must be running.
9. *Set the Failover Node*

   After you migrate the multiplex data, connect to the coordinator, and set the failover node.
10. *Troubleshooting Multiplex Migration*

    If you cannot migrate your multiplex database, try this alternate method.

Synchronizing the Multiplex Nodes

Check the SQL Remote and multiplex server log files for synchronization problems.

1. Start the multiplex server.
2. Start SQL Remote on all multiplex nodes.

   Give the multiplex time to propagate any changes throughout the multiplex. To do this, look at the write server console log file and check that the events starting with `ev_iqmpx` have successfully executed. By default, the server console log file is created in `$ASDIR/logfiles`.

   For example:

   ```
   Now accepting requestsOS Available: 933096K, Working Set: 83988K,
   Cache Target: 11483K
   OS Available: 860680K, Working Set: 83996K, Cache Target: 11483K
   Next time for 'ev_iqmpxq2w' is 2008/11/23 22:03:00.000
   Next time for 'ev_iqmpxstatus' is 2008/11/23 22:03:00.000
   OS Available: 859232K, Working Set: 84112K, Cache Target: 11489K
   OS Available: 861052K, Working Set: 84424K, Cache Target: 11489K
   OS Available: 860972K, Working Set: 84428K, Cache Target: 11489K
   OS Available: 850248K, Working Set: 85540K, Cache Target: 11579K
   OS Available: 850104K, Working Set: 85568K, Cache Target: 11579K
   Next time for 'ev_iqmpxq2w' is 2008/11/23 22:04:00.000
   Next time for 'ev_iqmpxstatus' is 2008/11/23 22:04:00.000
   OS Available: 850120K, Working Set: 85600K, Cache Target: 11579K
   Next time for 'ev_iqmpxq2w' is 2008/11/23 22:05:00.000
   Next time for 'ev_iqmpxstatus' is 2008/11/23 22:05:00.000
   OS Available: 852668K, Working Set: 85604K, Cache Target: 11579K
   ```

3. Wait for SQL Remote to scan the log files, then view the logs.

   Wait for SQL Remote to process any messages:

   ```
   I. 11/23 22:06:10. Scanning logs starting at offset 0001787252
   I. 11/23 22:06:10. Hovering at end of active log
   ```

4. Shut down SQL Remote and multiplex servers.

   If you simply shut down the multiplex servers, the SQL Remote servers detect that the multiplex servers are no longer running and shut themselves down. By default, the SQL Remote servers should shut themselves down within 60 seconds.

5. Shut down query servers in the multiplex. They are no longer required.
6. If the logs report no errors, verify the database.

Migrate Local Stores

To move the 12.7 local stores before migration, use **iqlsunload**.

To migrate your local store, consolidate node-specific information into either the existing 12.7 writer or the new SAP Sybase IQ 16.0 main store. Customize the process to meet your data requirements.

For query nodes with node- or department-specific information, use tablespaces and partitioning to achieve the same results.

If information is duplicated across your query nodes, you may need to migrate only a single query server's local store. The duplicated information on the other query servers becomes redundant and can be ignored for multiplex migration.

See also

- *Start the Multiplex Write Server* on page 80

Partitioning Query Server Data

If the same table exists on multiple query nodes, and each node has its own subset of the data, manually edit the local store migration scripts.

For a department-specific `employee` table on each query server, follow these basic steps:

1. Unload the schema and data from the query nodes.

   The `reload_schema.sql` script produced for each query node contains the same schema definition for `employee`.
2. Execute the `reload_schema.sql` from one of the query nodes against either the existing 12.7 writer or the new SAP Sybase IQ 16.0 main store.
3. Execute the `reload_data.sql` script from each of the query nodes against the same server.

   This procedure creates the `employee` table once but loads each query node data set.

See also

- *Addressing Overlapping Query Server Data* on page 76
- *Moving Local Stores* on page 79

Addressing Overlapping Query Server Data

If the same table exists on each query node with overlapping data sets, you must resolve the issue. Extract the data files to ensure that data sets are unique, or rename the tables and then reload all the unique tables.

1. Run **iqlsunload** against all query servers with local stores that have the data to consolidate.
2. Modify the `reload_schema.sql` and `reload_data.sql` files to use the new table names. Do not modify `extract_data.sql;` it references the table found in the query server's local store.
3. Run `extract_data.sql` from each node.

The following example shows modifications to the **iqlsunload** output to carry out step 2. Suppose that the `reload_schema.sql` script contains:

```
CREATE TABLE "DBA"."sales_order"
```

```
    "id"                unsigned int NOT NULL  IQ UNIQUE (648),
    "cust_id"               unsigned int NOT NULL  IQ UNIQUE (111),
    "order_date"            "datetime" NOT NULL  IQ UNIQUE (376),
    "fin_code_id"           char(2) NULL  IQ UNIQUE (1),
    "region"                char(7) NULL  IQ UNIQUE (5),
    "sales_rep"             unsigned int NOT NULL  IQ UNIQUE (75),
PRIMARY KEY ("id"),
```

Modify `reload_schema.sql` to:

```
CREATE TABLE "DBA"."q1_sales_order"

    "id"                unsigned int NOT NULL  IQ UNIQUE (648),
    "cust_id"               unsigned int NOT NULL  IQ UNIQUE (111),
    "order_date"            "datetime" NOT NULL  IQ UNIQUE (376),
    "fin_code_id"           char(2) NULL  IQ UNIQUE (1),
    "region"                char(7) NULL  IQ UNIQUE (5),
    "sales_rep"             unsigned int NOT NULL  IQ UNIQUE (75),
    PRIMARY KEY ("id"),
```

`extract_data.sql` contains:

```
---- Extract Table Data for table sales_order
-- NOTE: Approximately 57672 bytes of storage space.
-- will be required to extract the data for this table.
--
-- The following will unload the data for table
sales_order, row group 1, column group 1
```

```
SET TEMPORARY OPTION temp_extract_name1 =
'DBA_sales_order_1_1_DATA_1.inp';
SET TEMPORARY OPTION temp_extract_name2 =
'DBA_sales_order_1_1_DATA_2.inp';
SET TEMPORARY OPTION temp_extract_name3 =
'DBA_sales_order_1_1_DATA_3.inp';
SET TEMPORARY OPTION temp_extract_name4 =
'DBA_sales_order_1_1_DATA_4.inp';
SET TEMPORARY OPTION temp_extract_name5 =
'DBA_sales_order_1_1_DATA_5.inp';
SET TEMPORARY OPTION temp_extract_name6 =
'DBA_sales_order_1_1_DATA_6.inp';
SET TEMPORARY OPTION temp_extract_name7 =
'DBA_sales_order_1_1_DATA_7.inp';
SET TEMPORARY OPTION temp_extract_name8 =
'DBA_sales_order_1_1_DATA_8.inp';

SELECT id, cust_id, order_date,
IFNULL(fin_code_id, @null_string, fin_code_id),
IFNULL(region, @null_string, region), sales_rep
FROM "DBA"."sales_order"
WHERE rowid( "sales_order" ) >= 1
AND  rowid( "sales_order" ) <= 648;
```

```

SET TEMPORARY OPTION temp_extract_name1 = '';
SET TEMPORARY OPTION temp_extract_name2 = '';
SET TEMPORARY OPTION temp_extract_name3 = '';
SET TEMPORARY OPTION temp_extract_name4 = '';
SET TEMPORARY OPTION temp_extract_name5 = '';
```

Leave `extract_data.sql` code unchanged to extract the `sales_order` table from the query server.

Suppose that `reload_data.sql` contains:

```
-- Reload Table Data for table "sales_order"
-------------------------------------------------
ALTER TABLE "DBA"."sales_order" MODIFY cust_id NULL;
ALTER TABLE "DBA"."sales_order" MODIFY order_date NULL;
ALTER TABLE "DBA"."sales_order" MODIFY sales_rep NULL;

SET @max_row_id =
( SELECT MAX( rowid( "sales_order" ) )+1
FROM "DBA"."sales_order" );
```

```
SET @load_statement =
'LOAD TABLE "DBA"."sales_order"
(id, cust_id, order_date, fin_code_id NULL(
'''||@null_string||''' ) , region NULL(
'''||@null_string||''' ) , sales_rep)
FROM
'''||@extract_directory||'DBA_sales_order_1_1_DATA_1.
inp'',
'''||@extract_directory||'DBA_sales_order_1_1_DATA_2.
inp'', '''||@extract_directory||'DBA_sales_order_1_1_DATA_3.
inp'',
'''||@extract_directory||'DBA_sales_order_1_1_DATA_4.
inp'',
'''||@extract_directory||'DBA_sales_order_1_1_DATA_5.
inp'', '''||@extract_directory||'DBA_sales_order_1_1_DATA_6.
inp'', '''||@extract_directory||'DBA_sales_order_1_1_DATA_7.
inp'', '''||@extract_directory||'DBA_sales_order_1_1_DATA_8.
inp'' ROW DELIMITED BY ''\n'' QUOTES ON
ESCAPES OFF DEFAULTS OFF FORMAT ASCII
IGNORE CONSTRAINT ALL 0 START ROW ID
'||@max_row_id;
```

```
CALL IqExecuteCommand( @load_statement );
ALTER TABLE "DBA"."sales_order" MODIFY cust_id NOT
NULL;
ALTER TABLE "DBA"."sales_order" MODIFY order_date NOT
NULL;
ALTER TABLE "DBA"."sales_order" MODIFY sales_rep NOT
NULL;
```

Change `reload_data.sql` to:

```
-- Reload Table Data for table
"q1_sales_order"
-------------------------------------------------
ALTER TABLE "DBA"."q1_sales_order" MODIFY cust_id NULL;
ALTER TABLE "DBA"."q1_sales_order" MODIFY order_date
NULL;
```

```
ALTER TABLE "DBA"."q1_sales_order" MODIFY sales_rep
NULL;

SET @max_row_id = ( SELECT MAX( rowid( "q1_sales_order"
) )+1 FROM "DBA"."q1_sales_order" );
```

```
SET @load_statement =
'LOAD TABLE "DBA"."q1_sales_order"
(id, cust_id, order_date, fin_code_id NULL(
'''||@null_string||''' ) , region NULL(
'''||@null_string||''' ) , sales_rep) FROM
'''||@extract_directory||'DBA_q1_sales_order_1_1_DATA_
1.inp'',
'''||@extract_directory||'DBA_q1_sales_order_1_1_DATA_
2.inp'',
'''||@extract_directory||'DBA_q1_sales_order_1_1_DATA_
3.inp'',
'''||@extract_directory||'DBA_q1_sales_order_1_1_DATA_
4.inp'', '''||
@extract_directory||'DBA_q1_sales_order_1_1_DATA_5.inp'', '''||
@extract_directory||'DBA_q1_sales_order_1_1_DATA_
6.inp'', '''||@extract_directory||'DBA_q1_sales_order_1_1_DATA_
7.inp'', '''||@extract_directory||'DBA_q1_sales_order_1_1_DATA_
8.inp'' ROW DELIMITED BY ''\n'' QUOTES ON ESCAPES OFF
DEFAULTS OFF FORMAT ASCII IGNORE CONSTRAINT ALL 0
START ROW ID '||@max_row_id;
```

```
CALL IqExecuteCommand( @load_statement );ALTER TABLE
"DBA"."q1_sales_order" MODIFY cust_id NOT
NULL;
ALTER TABLE "DBA"."q1_sales_order" MODIFY order_date
NOT NULL;
ALTER TABLE "DBA"."q1_sales_order" MODIFY sales_rep NOT
NULL;
```

This example shows query server schema and data that require intervention during migration. Your situation may vary, but you have complete control of the content of the final `reload_schema.sql` and `reload_data.sql` files.

See also

- *Partitioning Query Server Data* on page 76
- *Moving Local Stores* on page 79

Moving Local Stores

Unload and move the 12.7 local stores.

Prerequisites

Upgrade to SAP Sybase IQ 12.7 ESD #5 or later.

Task

1. Source the `ASIQ-12_7.sh` or `.csh` file.
2. Run the 12.7 **iqlsunload** utility against each query server with a local store.
3. Edit `reload_schema.sql`:
    - Delete unwanted objects.
    - Change any commented objects in the `reload_schema.sql` that you want to reload.
    - Add commands to define any objects that you defined in **sp_mpxcfg_**`<servername>` procedures.
4. Edit `extract_data.sql` to remove objects you do not want to migrate. These objects are generally the same ones you removed from `reload_schema.sql`.
5. Use Interactive SQL to run `extract_data.sql` from your 12.7 local store.

    You now have unloaded the schema and data for local objects in the 12.7 local store.
6. Run the `reload_schema.sql` and `reload_data.sql` scripts against the 12.7 write server.

    Note: If you prefer, wait until the write server has been migrated to version 16.0 and run `reload_schema.sql` and `reload_data.sql` against the new coordinator.

See also

- *Partitioning Query Server Data* on page 76
- *Addressing Overlapping Query Server Data* on page 76

Start the Multiplex Write Server

To clean the internal state information, start the write server in single-node mode.

Note: You must specify your login and password as arguments to the start_server script.

Start the writer node with the server arguments **-gm 1** and **-iqmpx_sn 1**:

```
-gm 1 -iqmpx_sn 1
```

If you use administrative startup scripts, create a copy of the `start_server.sh` script to start the write server you want to migrate. For example, copy the existing file `start_server.sh` to a new file called `start_server_single_node.sh`.

Suppose that `start_server.sh` contains this startup command:

```
start_asiq -STARTDIR /work/iq-127/mpx/main @/work/iq-
127/mpx/main/params.cfg -n mpx_main $readonly $nomain -
x tcpip{port=62631} /work/iq-127/mpx/main/main.db
$dbkey
```

Add the two single node startup arguments to change the preceding command as follows in `start_server_single_node.sh`:

```
start_asiq -STARTDIR /work/iq-127/mpx/main @/work/iq-
127/mpx/main/params.cfg -n mpx_main -gm 1 -iqmpx_sn 1
$readonly $nomain -x tcpip{port=62631} /work/iq-127/
mpx/main/main.db $dbkey
```

There are now two script files, `start_server.sh` and `start_server_single_node.sh` to make the server ready for migration:

1. Start the writer node with `start_server_single_node.sh`.
2. Shut down the writer node.
3. Start the writer node with `start_server.sh`.
4. Shut down the writer node.
5. Shut down the SAP Sybase IQ 12.7 server.

See also

- *Migrate Local Stores* on page 75

Multiplex Migration Parameters

To migrate the multiplex, run **iqunload** with the appropriate parameters.

Minimum required parameters for a multiplex writer are **-au** (migrate database), **-c** (connection parameters), **ENG=** connection parameter and **-ms_filename**. The **ENG=** value must match the existing server name in SAP Sybase IQ 12.7, and the **-ms_filename** specifies the new main store for the migrated writer. This path must be the same for all nodes in the multiplex.

There are two differences in the way you will execute the **iqunload** utility for multiplex:

- Specify the engine name in the **-c** connection parameters. This is the same name that your `<mpx_dir>/<writer_node>/start_server` script file uses to start the writer node. The **iqunload** utility initially attempts to start the database server as simplex. This start requires the name of the server to match the naming conventions for the multiplex nodes. Once **iqunload** detects that the server is a multiplex node, it shuts the node down and restarts it using the **-iqmpx_sn 1** option.
- The name of the new main store must be visible and accessible by all nodes of the multiplex. This is important because the main store file name defaults to `new_system_main.iq`, and its location is relative to the catalog database file (.db). Later, when you synchronize the SAP Sybase IQ 16.0 multiplex, the catalog is replicated to the secondary nodes, formerly known as the query nodes. If you leave the default value for the main store name unchanged, the path remains `new_system_main.iq` and secondary nodes cannot find the shared main store.

For multiplex writers, required arguments are:

- **ENG** – argument specifies the multiplex main engine name. **iqunload** attempts to start the database and determine whether the database is a simplex or multiplex. If multiplex, the server name is enforced. If you are unsure of the server name, check the administrative script `start_server` in the database directory.

- **DBF** – argument must specify the actual path used to create the multiplex. If you are unsure of this, look at the `SYSIQFILE` table in your 12.7 server to verify the database path.
- **-ms_filename** – argument specifies the location of the new main store. This path must be visible and accessible by all servers in the multiplex.

For example:

```
iqunload -au -v -c
"uid=DBA;pwd=SQL;dbf=/sunx5prod/users/marshall/mpx127/
w1/w1.db;eng=w1_1234" -ms_filename
../shared/new_main_store.iq
```

```
 Sybase IQ Unload Utility Version 15.2.0.5533
Connecting and initializing
    2008-11-23 22:32:07 Unloading user and group
        definitions
    2008-11-23 22:32:08 Unloading table definitions
    2008-11-23 22:32:09 Unloading index definitions
    2008-11-23 22:32:09 Unloading functions
    2008-11-23 22:32:09 Unloading view definitions
    2008-11-23 22:32:09 Unloading procedures
    2008-11-23 22:32:09 Unloading triggers
    2008-11-23 22:32:09 Unloading SQL Remote         definitions
    2008-11-23 22:32:09 Unloading MobiLink definitions
    2008-11-23 22:32:10 Creating new database
    2008-11-23 22:32:48 Reloading user and group
       definitions   2008-11-23 22:32:48 Reloading table definitions
    2008-11-23 22:32:53 Reloading index definitions
    2008-11-23 22:32:53 Reloading functions
    2008-11-23 22:32:53 Reloading view definitions
    2008-11-23 22:32:53 Reloading procedures
    2008-11-23 22:32:53 Reloading triggers
    2008-11-23 22:32:53 Reloading SQL Remote
        definitions
    2008-11-23 22:32:53 Reloading MobiLink definitions
```

```
Successfully backed up file "/sunx5prod/users/marshall/mpx127/w1/
w1.db" by
renaming it to
"/sunx5prod/users/marshall/mpx127/w1/w1.db.before_schema_reload".
Successfully backed up file
"/sunx5prod/users/marshall/mpx127/main.db" by renaming it to
"/sunx5prod/users/marshall/mpx127/main.db.before_schema_reload".
Successfully backed up file
/sunx5prod/users/marshall/mpx127/main.iqmsg" by renaming it to
"/sunx5prod/users/marshall/mpx127/main.iqmsg.before_schema_reload".
Successfully reloaded schema of database
"/sunx5prod/users/marshall/mpx127/main.db".
```

Verifying the Migrated Multiplex Database

Verify the migrated database in read-only mode and correct any errors.

1. Start the database using the read-only switch, **-iqro 1**. Start the coordinator (the 12.7 write server) using both **-iqro 1** and single node mode, **-iqmpx_sn 1**.

   When starting the coordinator in 16.0, use the same port used by the 12.7 writer server.
2. Issue a **CHECKPOINT** command.
3. Run **sp_iqcheckdb** in verify mode:

   ```
   sp_iqcheckdb 'verify database'
   ```
4. Issue a **COMMIT** statement.

The server is currently in read-only mode, and cannot complete some post migration tasks. Additionally, the verification reports some problems with Block Count Mismatch, Blocks Leaked, and Unallocated Blocks in Use. No other segments of the verify database should report any errors.

For example:

```
'** Block Count Mismatch','79','*****'
'** Blocks Leaked','25','*****'
'** Unallocated Blocks in Use','104','*****'
```

Examine the **sp_iqcheckdb** report for errors. If you need to contact SAP Sybase Technical Support, you must provide the output from **sp_iqcheckdb**.

Starting the Coordinator

Starting the multiplex coordinator in single-node mode (**-iqmpx_sn**) and read-only (**-iqro**) performs some initial database checks. For coordinators, the server must reset an identity cookie before you can use the multiplex.

Once you successfully restart the coordinator with **iqro 1** and **iqmpx_sn 1**, shut it down and restart it without any special switches.

For example:

```
start_iq @params.cfg -n mpx_main -iqmpx_ov 1 -x 'tcpip{port=62631}' /
workserver/work/iq-127/mpx/main.db
```

Manually Synchronize the Secondary Nodes

To start the secondary nodes, install SAP Sybase IQ 16.0, then synchronize from the coordinator node. When you migrate a query node, it becomes a reader node.

1. Back up the query node files. Back up existing catalog `.db`, catalog `.log` and `iqmsg` files.

   For example:

   ```
   rename /sunx5prod/users/work/iq-127/mpx/q1/q1.db
   /sunx5prod/users/work/iq-127/mpx/q1/q1.db.before_schema_reload
   ```

```
rename /sunx5prod/users/work/iq-127/mpx/q1/q1.log /sunx5prod/
users/work/iq-127/mpx/q1/q1.log.before_schema_reload
rename /sunx5prod/users/work/iq-127/mpx/q1/q1.iqmsg
/sunx5prod/users/work/iq-127/mpx/q1/q1.iqmsg.before_schema_reload
```

2. Issue a **dbbackup** command to synchronize servers. You might have a different name for the query node's catalog file, depending on your configuration. In the following example, `q1.db` is the catalog file name on the query node:

```
dbbackup -y -x -c
"uid=dba;pwd=sql;eng=mpx_main;dbf=/sunx5prod/users/
work/iq-127/mpx/main/main.db"
/sunx5prod/users/work/iq-127/mpx/q1
```

```
SQL Anywhere Backup Utility Version 11.0.1.5533 Debug
(702 of 699 pages, 100% complete)
Transaction log truncated
Database backup completed
```

3. If your query nodes do not use a different catalog database name, skip to step 4.

   Step 2 synchronizes the catalog database file from the coordinator. If you prefer to use the same catalog database file name as the coordinator, adjust any server start and stop administration scripts on the secondary nodes to use the new name.

   To retain the same catalog database file names:

   - Rename the synchronized coordinator catalog database file name. For example, assuming the coordinator file was called `main.db` and the secondary server was called `q1.db`, enter:

   ```
   mv main.db q1.db
   ```

   ```
   rename main.db q1.db
   ```

   - Rename the log file for the query node. This is necessary as the file renamed above still contains an internal pointer to `main.log`:

   ```
   dblog -t q1.log q1.db
   ```

4. Start the secondary server in normal mode:

```
start_iq @params.cfg -n mpx_q1 -x
'tcpip{port=62632}' -o /worksrver/work/
iq-127/mpx/q1/o.out -Z -zr all -zo /worksrver/
iq-127/mpx/q1/zo.out /workserver/work/
iq-127/mpx/q1/main.db
```

   The above command line is derived from your existing query server `start_server` administration script.

5. Repeat these steps on the remaining secondary nodes that you want to migrate.

Start the Secondary Nodes

To start the secondary nodes, all nodes of the multiplex must be running.
Start the secondary servers with the command line startup utility.

For example:

```
start_iq @params.cfg -n <server_name> database_file.db
```

Where <*server_name*> specifies the secondary server. You can obtain the name from the existing start server administration script. The specified `database_file.db` is the name resulting after you performed the secondary node synchronization.

Set the Failover Node

After you migrate the multiplex data, connect to the coordinator, and set the failover node. Use a command like this to set the failover node:.

```
ALTER MULTIPLEX SERVER servername ASSIGN AS FAILOVER SERVER
```

Where *servername* is one of the secondary nodes.

Troubleshooting Multiplex Migration

If you cannot migrate your multiplex database, try this alternate method.

- Drop all query nodes, to change the SAP Sybase IQ 12.7 multiplex to a simplex database.
- Follow the steps for simplex databases to migrate the database to SAP Sybase IQ 16.0.
- Convert the simplex SAP Sybase IQ 16.0 database to multiplex, following the steps in *Administration: Multiplex > Create Multiplex Servers > Converting Databases to Multiplex*.

Postmigration Tasks

SAP Sybase IQ 16 databases upgraded from 12.7 are initially set to run in 15.x compatibility mode. To complete the change from 15.x to 16.0, you must explicitly change several 15.x compatibility settings to complete the 16.0 upgrade.

Indexes

- In Fast Projection (`FP`) indexes, continuous `NBit` dictionary compresion replaces `FP(1)`, `FP(2)`, and `FP(3)` byte dictionary compression. `FP(1)`, `FP(2)`, and `FP(3)` indexes roll over to `NBit(8)`,`NBit(16)`, and `NBit(24)` respectively. All data types except LOB (both character and binary) and BIT data types may be `NBit` columns.

  If `FP_NBIT_IQ15_COMPATIBILITY` is OFF, `IQ UNIQUE` determines whether the column loads as `Flat FP` or `NBit`. Setting `IQ UNIQUE` to 0 loads the column as `Flat FP`. Columns without an `IQ UNIQUE` constraint load as `NBit` up to the `NBit` auto-sizing limits.
- New tiered `HG` index structure decouples load performance from `HG` index size. In 15.x, load throughput could degrade as the amount of data in an `HG` index increased. As the index grew, loading the same amount of data could take more time. The new tiered structure decouples load performance from the `HG` index size to increase throughput.

  The `CREATE_HG_WITH_EXACT_DISTINCTS` option determines whether newly created `HG` indexes are tiered or non-tiered. If this option is ON, all new `HG` indexes are non-tiered. To take advantage of the new structure, set this option to OFF. Use **sp_iqrebuildindex** to convert non-tiered HG indexes to tiered HG and vice-versa.

Constraints

Constraint	Description
IQ UNIQUE	In SAP Sybase IQ 16.0, IQ UNIQUE explicitly defines the expected cardinality of a column and determines whether the column loads as Flat FP or NBit. Columns retain their IQ UNIQUE(*n*) value during a 15.x to 16.0 database upgrade. Setting IQ UNIQUE to 0 loads the column as Flat FP. Columns without an IQ UNIQUE constraint or columns with an IQ UNIQUE *n* value that is less than the limit defined by the FP_NBIT_AUTOSIZE_LIMIT option is not necessary. Auto-size functionality automatically sizes all low or medium cardinality columns as NBit. Use IQ UNIQUE in cases where you want to where you want to load the column as Flat FP or when you want to load as NBit and the number of distinct values exceeds the auto-size limits.

Options

Option	Description
FP_NBIT_IQ15_COMPATIBILITY	Provides tokenized FP support similar to that available in 15.x. This option is ON by default in all 16.0 databases upgraded from 15.x and OFF in all newly created 16.0 databases. • If this option is ON, the database engine uses the MINIMIZE_STORAGE, FP_LOOKUP_SIZE, and FP_LOOKUP_SIZE_PPM options to optimize column storage. These options are ignored in 16.0. • If this option is OFF, the database engine ignores 15.x options and columns conform to SAP Sybase IQ NBit storage options. Set this option to OFF to take advantage of NBit column compression.

Option	Description
CREATE_HG_WITH_EXACT_DISTINCTS	Determines whether new `HG` indexes explicitly created with a **CREATE INDEX** command, or implicitly creating or altering a table with a PRIMARY KEY or a FOREIGN KEY declaration, are tiered or non-tiered. This option is ON in 16.0 databases upgraded from 15.x and all newly created 16.0 databases. If this option is ON, all new `HG` indexes are non-tiered. To take advantage of the new tiered `HG` index structure, set this option to OFF. Use **sp_iqrebuildindex** to convert non-tiered `HG` indexes to tiered `HG` and vice-versa.
CREATE_HG_AND_FORCE_PHYSICAL_DELETE	Governs 16.0 delete behavior for tiered `HG` indexes. This option determines whether SAP Sybase IQ performs a physical delete immediately or defers the delete to a point later in the load. CREATE_HG_AND_FORCE_PHYSICAL_DELETE is ON by default, which instructs SAP Sybase IQ to perform physical deletes.
REVERT_TO_V15_OPTIMIZER	**REVERT_TO_V15_OPTIMIZER** forces the query optimizer to mimic SAP Sybase IQ 15.x behavior. **REVERT_TO_V15_OPTIMIZER**='ON' by default in all 16.0 databases upgraded from 15.x. **REVERT_TO_V15_OPTIMIZER**='OFF' by default in all newly created SAP Sybase IQ 16.0 databases. If you plan to use SAP Sybase IQ hash partitioning features, set the **REVERT_TO_V15_OPTIMIZER** ='OFF' in databases upgraded from 15.x to 16.0.

Object Names

Reserved words cannot be used as object names.

An SAP Sybase IQ 15.x database could contain tables, columns, and other objects named row. In SAP Sybase IQ 16.0, row is a reserved word and cannot be used as an object name.

To use a reserved word as an object name, enclosed the object name in brackets (regardless of the `QUOTED_IDENTIFIER` setting) or double quotes (if `QUOTED_IDENTIFIER='ON'` [default]):

Stored Procedures

Use these stored procedures to review and change column indexes and constraints:

Procedure	Description
sp_iqcolumnmetadata	Returns index metadata for all columns in one or more tables.
sp_iqindexmetadata	Returns details about column indexes, including the index types (`Flat FP`, `NBit`, `HG`, and tiered `HG`), distinct counts, `IQ UNIQUE` *`n`* value, and `NBit` dictionary size.
sp_iqrebuildindex	Rebuilds `FP` indexes (`Flat FP` as `NBit`, or `NBit` as `Flat FP`) and `HG` indexes (single `HG` as tiered `HG`, or tiered `HG` as single `HG`). Before you can insert or update new data, you must rebuild all columns greater than 255 bytes wide. The `index_clause` can reset `IQ UNIQUE` *`n`* to an explicit value from 0 (to recast an `NBit` column to `Flat FP`) up to the limits defined in the **FP_NBIT_AUTOSIZE_LIMIT** and **FP_NBIT_LOOKUP_MB** options. **sp_iqrebuildindex** also enables read-write access to columns that contain large object (LOB) data. LOB columns migrated from 15.x databases are read-only until you run **sp_iqrebuildindex**. The estimated cardinality for `NBit` columns with an `IQ UNIQUE` value below or equal to the **FP_NBIT_AUTOSIZE_LIMIT** is stored as 0 regardless of the **FP_NBIT_IQ15_COMPATIBILITY** setting. This affects the value returned from **sp_iqindexmetadata**.
sp_iqindexrebuildwidedata	Identifies wide columns that you must rebuild before they are available for read/write activities. **sp_iqindexrebuildwidedata** also generates a list of statements that you can use to rebuild the columns. This applies to CHAR, VARCHAR, BINARY, and VARBINARY columns wider than > 255 characters, as well as all Long Varchar and Long Binary columns.

Re-create Indexes for EUC_TAIWAN Data

In SAP Sybase IQ 15 and later, the character encoding specification for the EUC–TAIWAN collation now uses the EUC_TW character set. You must re-create indexes on data in version 12.7 or earlier databases that use the EUC_TAIWAN collation to make them work with SAP Sybase IQ 16.

Update Configuration Files

Compare your existing `params.cfg` files with the new `default.cfg` file created by the installation. The installation does not update or overwrite existing `params.cfg` files. In each `params.cfg` file, update any parameter defaults that differ from those in the `default.cfg` file, while maintaining any customized parameter settings that are appropriate for your system. Add any new startup parameters in `default.cfg` to your `params.cfg` file. The **-gl** parameter, for example, is required for server startup in version 12.5 and later.

Preserve Database Options

SAP Sybase IQ preserves the settings of all 12.7 database options that are still valid in migrated databases. Check for deprecated features.

Back Up Your Databases

- Back up your databases again with the **BACKUP** statement. If you use the **BACKUP** statement instead of a system–level backup, you can run backups and queries concurrently.
- For a multiplex migration, back up only the coordinator only in this manner. For secondary servers, run the **dbbackup** utility from the secondary server directory.

Additional Information

- *Administration: Database > Index SAP Sybase IQ Columns > Index Types Comparison > Fast Projection (FP) Index*
- *Administration: Database > Index SAP Sybase IQ Columns > Index Types Comparison > High_Group (HG) Index*
- *Reference: Statements and Options > SQL Statements > ALTER TABLE*
- *Reference: Statements and Options > Database Options > Alphabetical List of Options > FP_NBIT_IQ15_ COMPATIBILITY_MODE*
- *Reference: Statements and Options > Database Options > Alphabetical List of Options > CREATE_HG_WITH_EXACT_DISTINCTS*
- *Reference: Building Blocks, Tables, and Procedures > System Procedures > Alphabetical List of System Stored Procedures > sp_iqindexmetadata*
- *Reference: Building Blocks, Tables, and Procedures > System Procedures > Alphabetical List of System Stored Procedures > sp_iqrebuildindex*

Hardware Changes

Perform these steps to move your software to a new hardware platform.

Moving 32-Bit Databases to 64-bit Platforms

Perform these steps to move a 32-bit database to a 64-bit platform.

Prerequisites

- Review backup and restore procedures:
  - For simplex servers, see *Administration: Backup, Restore, and Data Recovery* .
  - For multiplex servers, see *Administration: Multiplex > Back Up and Restore*.
- Make note of the 32-bit server raw device and IQ store path names. Raw device and IQ store path names on the 64-bit target must match those on the 32-bit machine.

Task

1. Log in to your 32-bit server and back up the database.
2. Copy the backup to the 64-bit machine, and restore the database.

   You may need to rename raw device and path names to ensure they match. See *Administration: Multiplex > Back Up and Restore*.
3. On the 64-bit machine, do one of the following:

Server	Action
Simplex	Start the database with the appropriate startup flags.
Multiplex	Start the coordinator with the **-iqmpx_sn 1**, **-gm 1** , **-iqro 1**, and **-iqmpx_ov 1** startup flags.

4. Start Interactive SQL and connect to the database.
5. Use **DROP MULTIPLEX SERVER** to drop all existing secondary nodes.
6. Use **ALTER DATABASE UPGRADE** to upgrade the database.

   See *Reference: Statements and Options > SQL Statements > ALTER DATABASE Statement*.
7. Run **sp_iqcheckdb ('allocation database')** and verify that the database is error free.

   sp_iqcheckdb checks the validity of the current database. See *Reference: Building Blocks, Tables, and Procedures > System Procedures > sp_iqcheckdb Procedure*.
8. Perform these steps for multiplex servers only:

a) Shut down and restart the coordinator normally (without the **-gm 1** , **-iqro 1**, and **-iqmpx_ov** startup flags).

b) Use **CREATE MULTIPLEX SERVER STATEMENT** to recreate the secondary nodes.

See also

- *Converting to a New Hardware Platform* on page 91

Converting to a New Hardware Platform

Perform these steps to move a database to another platform with the same endian structure.

Platforms must share the same endian structure. Move your database, then migrate your data.

1. Back up the database.
2. Shut down the SAP Sybase IQ server.
3. Install the server on the new platform. Your migration can take place on the same or a different machine.
4. Start the server on the new hardware platform.
5. Connect to the utility database, `utility_db`.
6. Restore the database from the backup you created in step 1.
7. Shut down the server and restart it against the restored database.
8. Start Interactive SQL and run **ALTER DATABASE UPGRADE**.

Note: If the SAP Sybase IQ version is more recent than the version on legacy platform, you must upgrade the database.

See also

- *Moving 32-Bit Databases to 64-bit Platforms* on page 90

Index

B

backwards compatible 57

C

Cache Settings
- Extremely Large Schemas 67
- Increasing 67
- Large Schemas 67

compatibility role
- delete 54
- drop 54
- migrate 53
- re-create 55
- revoke 53

compatibility roles 21
- grant 31
- SYS_AUTH_BACKUP_ROLE 39
- SYS_AUTH_DBA_ROLE 34
- SYS_AUTH_MULTIPLEX_ADMIN_ROLE 40
- SYS_AUTH_OPERATOR_ROLE 41
- SYS_AUTH_PERMS_ROLE 42
- SYS_AUTH_PROFILE_ROLE 44
- SYS_AUTH_READCLIENTFILE_ROLE 45
- SYS_AUTH_READFILE_ROLE 44
- SYS_AUTH_RESOURCE_ROLE 46
- SYS_AUTH_SA_ROLE 31
- SYS_AUTH_SPACE_ADMIN_ROLE 48
- SYS_AUTH_SSO_ROLE 33
- SYS_AUTH_USER_ROLE 49
- SYS_AUTH_VALIDATE_ROLE 50
- SYS_AUTH_WRITEFILE_ROLE 51
- SYS_AUTH_WRITEFILECLIENT_ROLE 52

Converting to a New Hardware Platform 91

Coordinator
- IQ 12.7 Multiplex Migration 83
- Starting 83

D

Database Migration
- IQ 12.7 57

database upgrade
- Sybase IQ 15 multiplex 8

Database Upgrades
- IQ 15 database upgrades 15
- IQ 16 database upgrades 6, 14
- IQ 16.0 SPx database upgrades 6, 14
- multiplex 6, 14, 15
- Preliminary Steps 4, 12
- SAP Control Center 15
- SCC 15
- security model updates 6, 14, 15
- simplex 6, 14, 15
- sp_iqcheckdb 6, 14, 15

E

EUC_TAIWAN Data 85

Examples
- iqlsunload 61, 64
- iqunload 58

F

Failover Node
- Setting 85

H

Hardware Changes 90
- Converting to new hardware platform 91
- Moving 32-bit databases 90

I

IQ 12.7 72
- Database Migration 57

IQ 12.7 Cache Settings
- Extremely Large Schemas 67
- Increasing 67
- Large Schemas 67

IQ 12.7 Data Storage Changes
- dbspaces 69
- Main Store 69
- Migrating IQ_SYSTEM_MAIN 69

IQ 12.7 Database Migration 65
- Database Migration 57
- Migrating Legacy Databases 72
- Migration Issues 64
- Migration Utilities 58
- Postmigration Tasks 85
- Preliminary Steps 57
- Simplex Database Migration 72
- Unloading Legacy Schemas 71

IQ 12.7 local stores
- migrating 75
- moving 75

IQ 12.7 Migration
- Migrating Legacy Databases 72
- Migration Utilities 58
- Postmigration Tasks 85
- Preliminary Steps 57
- Simplex Database Migration 72

IQ 12.7 Migration Issues 64, 70
- Data Storage Changes 69
- invalid database names 65, 66
- invalid table names 65, 66
- invalid user names 65, 66
- join indexes 65, 66
- Output Logs 68
- Post Migration Files 70
- Schema Size 67
- Syntax Changes 66
- unsupported objects 65
  - See also IQ 12.7 Database Migration

IQ 12.7 Migration Utilities 58
- iqlsunload 61
- iqunload 58
- Migration Issues 64

IQ 12.7 Multiplex Migration 74
- iqunload 81
- Migrate Local Stores 75
- Moving Local Stores 79
- Multiplex Migration Parameters 81
- Multiplex Write Server 80
- Partitioning Query Server Data 76
- Setting Failover Node 85
- Starting Secondary Nodes 84
- Starting the Coordinator 83
- Synchronize and Shutdown Multiplex 75
- Synchronizing Secondary Nodes 83
- Troubleshooting 85
- Verify Migrated Database 83

IQ 12.7 Output Logs
- Engine Logs 68
- iqunload logging 68
- Obsolete Stored Procedures 68
- Server Not Found 68

IQ 12.7 Postmigration
- Postmigration Tasks 85

IQ 12.7 Postmigration Tasks
- Database Backup 85
- Database options, preserving 85
- Preserving options (postmigration) 85
- Recreate EUC_TAIWAN Data indexes 85
- Update Configuration Files 85

IQ 12.7 Schema Size
- Extremely Large Schemas 67
- Large Schemas 67

IQ 12.7 Simplex Database Migration
- database verification (sp_iqcheckdb) 72, 73
- Migrate the Legacy Database 72
- Verify the Migrated Database 73

IQ 12.7 Support Processes
- iqsrv16 64
- iqunlspt 64

IQ 12.7 Syntax Changes
- CREATE TRIGGER 66
- DECLARE LOCAL TEMPORARY TABLE 66
- OUT or INOUT paramters 66
- Outer Joins 66
- Reserved Words 66

iqlsunload
- Examples 61
- Moving Local Stores 79
- Parameters 61
- Permissions 61
- Syntax 61
- Usage 61

iqsrv16
- default options 64

iqunload
- Examples 58
- Multiplex Migration Parameters 81
- Parameters 58
- Permissions 58
- schema unload mode 71
- Syntax 58
- Unloading Legacy Schemas 71
- Usage 58

iqunlspt
- cache settings 64
- default options 64

L

logical servers
- multiplex access 8
- names 8

M

Migrating Legacy Databases 72
- Postmigration Tasks 85

Migration
- Hardware Changes 90

Migration Mode
- iqunload 58

Moving 32-bit databases 90

Moving Local Stores
- IQ 12.7 Multiplex Migration 79

multiplex
- IQ 12.7 local stores 75

Multiplex Upgrades 6, 14

Multiplex Write Server
- IQ 12.7 Multiplex Migration 80

P

Parameters
- iqlsunload 61, 64
- iqunload 58

Partitioning Query Server Data
- IQ 12.7 Multiplex Migration 76

Permissions
- iqlsunload 61, 64
- iqunload 58

Post Migration Files
- See IQ 12.7 Migration Issues

Preliminary Steps
- Database Upgrades 4, 12
- IQ 12.7 database migration 57

R

REMOTE DBA
- Changes 29

role-based security model 20, 21
- system role 23
- user-extended roles 23

role-based syntax 57

S

schema unload mode
- iqunload 71
- Unloading Legacy Schemas 71

Schema Unload Mode
- iqunload 58

Secondary Nodes
- IQ 12.7 Multiplex Migration 83
- Starting 84
- Synchronizing 83

Simplex Database Migration 72

Simplex Upgrades 6, 14

sp_auth_sys_role_info 57

stored procedures 6, 14, 15, 66

super-user 24

Sybase IQ 12.6/12.7
- multiplex migration 74

Sybase IQ 12.7 Migration
- Migration Issues 64
- Unloading Legacy Schemas 71

Sybase IQ 15
- logical server 8
- login policy migration 8
- multiplex upgrade 8

Syntax
- iqlsunload 61
- iqunload 58

SYS_AUTH_BACKUP_ROLE
- grant 39
- system privileges granted 40

SYS_AUTH_DBA_ROLE
- grant 34
- roles granted 35
- system privileges granted 36

SYS_AUTH_MULTIPLEX_ADMIN_ROLE
- grant 40
- system privileges granted 41

SYS_AUTH_OPERATOR_ROLE
- grant 41
- system privileges granted 42

SYS_AUTH_PERMS_ADMIN_ROLE
- roles granted 43
- system privileges granted 43

SYS_AUTH_PERMS_ROLE
- grant 42

SYS_AUTH_PROFILE_ROLE
grant 44
SYS_AUTH_READCLIENTFILE_ROLE
grant 45
system privileges granted 46
SYS_AUTH_READFILE_ROLE
grant 44
system privileges granted 44, 45
SYS_AUTH_RESOURCE_ROLE
grant 46
system privileges granted 47
SYS_AUTH_SA_ROLE
grant 31
system privileges granted 31
SYS_AUTH_SPACE_ADMIN_ROLE
grant 48
SYS_AUTH_SPACE_ROLE
system privileges granted 49
SYS_AUTH_SSO_ROLE
grant 33
system privileges granted 34
SYS_AUTH_USER_ADMIN_ROLE
system privileges granted 49
SYS_AUTH_USER_ROLE
grant 49
SYS_AUTH_VALIDATE_ROLE
grant 50
system privileges granted 51
SYS_AUTH_WRITECLIENTFILE_ROLE
grant 52
SYS_AUTH_WRITEFILE_ROLE
grant 51
system privileges granted 52
SYS_AUTH_WRITEFILECLIENT_ROLE
system privileges granted 52
system procedures
changed behaviour 17
system role
migrating compatibility role 56

T

Transact-SQL outer joins 66

U

Unloading Legacy Schemas 71
Unsupported 12.7 Metadata
invalid database names 65
invalid table names 65
invalid user names 65
join indexes 65
Unsupported 12.7 Objects
AUTO logical server 65
invalid database names 65
invalid logical server names 65
invalid table names 65
invalid user names 65
join indexes 65
upgrade, database
Sybase IQ 15 multiplex 8
Usage
iqlsunload 61, 64
iqunload 58

V

Verify Migrated Database
Multiplex Migration 83