Administration Guide | PUBLIC

SAP IQ 16.0 SP 11

Document Version: 1.0 – 2019-01-15

Administration: User Management and Security

© 2020 SAP SE or an SAP affiliate company. All rights reserved.

THE BEST RUN

Content

1 Security Management

SAP® IQ provides a role-based security model for controlling access to database objects and executing privileged operations. This model provides complete control and granularity for the privileges you want to grant to users. Each privileged operation in a database requires one or more system or object-level privileges be assigned to the user to execute the operation.

A system privilege allows users to perform authorized database tasks. For example, assign the CREATE TABLE system privilege to a user to allow him or her to create self-owned tables.

An object-level privilege allows a user to perform an authorized task on a specified object. For example, assign ALTER object-level privilege on `TableA` to a user to allow him or her to alter that table, but no other tables.

A role is a container that may contain one or more system privileges, object-level privileges, and other roles. Granting a role to a user is equivalent to granting the user the underlying system and object-level privileges of the role.

All new users are automatically granted the PUBLIC system role, which gives them the ability to:

- View the data stored in the system views
- Execute most system stored procedures

Once you have created a new user, you can:

- Grant user-defined roles, system roles, system privileges, and object-level privileges to it.
- Assign a login policy it. By default, a user is assigned to the root login policy.
- Set it as the publisher or as a remote user of the database for use in a SQL Remote system.

Each new or migrated SAP IQ database includes a predefined set of roles you can use to get started. These system roles act as a starting point for implementing role-based security.

> i Note
>
> If you have used versions of SAP IQ earlier than 16.0, SAP recommends that you review the sections on how the security model has changed from the authority/permission/group model to the role/privilege/user-extended role model under *Upgrading to Role-Based Security* in the SAP IQ Installation and Update Guide appropriate to your operating system.

In this section:

Plan and Implement Role-Based Security [page 8]
There is a distinct workflow to planning and implementing a role-based security model.

Roles [page 9]
A role is a container that can contain system privileges, object-level privileges, and roles. Granting privileges to and revoking privileges from a role is the same as for a user. A role and user cannot have the same name.

Privileges [page 46]
A privilege grants users the ability to perform an authorized operation on the system. For example, altering a table is a privileged operation, depending on the type of alteration you are making.

[Passwords [page 126]]
A user can be granted the ability to manage other users' passwords. You can configure password management to require one or two users to complete a password change.

Impersonation [page 134]
A user can temporarily assume (impersonate) the specific roles and system privileges of another user to perform operations, provided he or she already has the minimum required privileges to perform the task to begin with.

Users [page 145]
User management includes the creation and deletion of user IDs, as well as password management.

Login Policies [page 156]
A login policy defines the rules that SAP IQ follows to establish user connections. Each login policy is associated with a set of options called login policy options.

User Connections [page 162]
There are several ways to manage user connections.

Security with Views and Procedures [page 167]
You can use views and stored procedures to tailor privileges to suit the needs of your enterprise.

Data Confidentiality [page 174]
You can secure communications between a client and the SAP IQ server, or between an SAP IQ client and the database server using Transport Layer Security (TLS).

Utility Database Server Security [page 187]
SAP IQ includes a phantom database, called the utility database, that has no physical representation, and which can contain no data.

Data Security [page 189]
Since databases may contain proprietary, confidential, or private information, it is important that you ensure that the database and the data in it are designed for security.

1.1 Plan and Implement Role-Based Security

There is a distinct workflow to planning and implementing a role-based security model.

Designing the Security Hierarchy

1. Identify the various authorized tasks to be performed by users. Group closely related tasks. Groupings can be based on any organizational structure—departmental, functional, and so on. You can create a role hierarchy that matches the organizational hierarchy. Assign a name to each grouping. These groupings represent the roles you create.
2. Identify the system privileges and object-level privileges required to perform each authorized task identified.
3. Identify the users to perform the various authorized tasks. Associate them with the applicable roles or with identified individual tasks.

4. (Optional) Identify administers for the roles you create. Administrators can grant and revoke the role to other users.
5. (Optional) Identify administrators for the system privileges and object-level privileges that are not part of the roles you create.

Build the Security Hierarchy

1. Create the required roles. See *Roles*.
2. To each role, grant the system privileges. See *Roles* and *Privileges*.
3. Create the users. See *Users*.
4. Grant applicable roles to each user, including administrative rights where applicable. See *Roles*.
5. Grant applicable object-level and system privileges to users (not already indirectly granted through roles), including administrative rights where applicable. See *Privileges*.

Related Information

Roles [page 9]
Privileges [page 46]
Users [page 145]

1.2 Roles

A role is a container that can contain system privileges, object-level privileges, and roles. Granting privileges to and revoking privileges from a role is the same as for a user. A role and user cannot have the same name.

In this section:

The `sp_displayroles` stored procedure returns all roles that are granted to the specified system privilege, system role, user-defined role, or user name, or displays the entire hierarchy tree of roles.

Determining the Roles and Privileges Granted to a User [page 46]
The `sp_has_role` stored function returns an integer value that indicates whether the invoker of the procedure has been granted the specified system privilege or user-defined role.

1.2.1 User-Defined Roles

A user-defined role is a custom collection of system and object-level privileges, typically created to group privileges that are related to a specific task or set of tasks.

A user-defined role:

- Can be a standalone object with no login privileges, which can own objects.
- Can be a database user with the ability to act as a role (user-extended role). If an existing user ID has login privileges, the user-extended role retains the login privileges.
- Can be granted privileges on other objects.
- Can be granted privileges of other roles.
- Has a case-insensitive name.

The granting of a user-defined role is semantically equivalent to individually granting each of its underlying system and object-level privileges.

You cannot convert a user-defined role to a user-extended role, and vice versa.

> i Note
>
> Unless otherwise noted, the term user-defined role refers to both user-extended and user-defined roles.

In this section:

Creating a User-Defined Role [page 11]
Create a new user-defined role.

Converting an Existing User to a User-Extended Role [page 13]
You can extend an existing user ID to act as a role. This is useful when you have a user who is assigned a set of system and object-level privileges that you want to grant to another user.

Converting a User-Extended Role Back to a User [page 14]
You can convert a user-extended role back to a regular user.

Adding a User-Defined Role to a User or Role [page 15]
Add membership in a user-defined role to a user or role (grantee), with or without administrative rights.

Removing Members from a User-Defined Role [page 17]
Remove a user or role as a member of a role. The user or role loses the ability to use any underlying system privileges or roles of a role, along with the ability to administer the role, if granted.

Deleting a User-Defined Role [page 18]
Delete a user-defined role from the database as long as all dependent roles retain the minimum required number of administrator users with active passwords. If the minimum value is not maintained, the command fails.

Role and Global Role Administrators [page 19]
Role administrators and global role administrators grant and revoke user-defined roles to users and other roles. You can add and remove role and global role administrators on a role as needed.

1.2.1.1 Creating a User-Defined Role

Create a new user-defined role.

Prerequisites

MANAGE ROLES system privilege.

Context

A user-defined role cannot have a login password. When creating a user-defined role, you can appoint administrators for the role, and indicate whether they are also to be members of the role. If you do not specify any administrators, the global role administrator (any user granted the MANAGE ROLES system privilege) becomes the default administrator of the role.

However, if at least one role administrator is specified, global role administrators cannot manage the role because the SYS_MANAGE_ROLES_ROLE system privilege is not automatically granted to the role with administrative rights. For this reason, SAP strongly recommends that you either do not define any role administrators when creating or converting a role (add them after creation), or explicitly grant the SYS_MANAGE_ROLES_ROLE system privilege with administrative rights only along with any role administrators during the process.

You can add or remove role administrators can be added and removed after creating a role. If you attempt to create a new role using an existing role name, the statement fails.

Procedure

To create a new user-defined role, execute one of these statements:

Create Condition	Statement
Global role administrator only; **no role administrators**	`CREATE ROLE <role_name>`

Role administrators with no role membership;	`CREATE ROLE <role_name>`
	`WITH ADMIN ONLY <admin_name [,...]>`

Create Condition	Statement
no global role administrator	
Role administrators with role membership; no global role administrator*	CREATE ROLE <role_name> WITH ADMIN <admin_name [,...]>
Role administrators with no role membership; with global role administrator*	CREATE ROLE <role_name> WITH ADMIN ONLY SYS_MANAGE_ROLES_ROLE, <admin_name [,...]>

*Since global role administrators cannot be granted membership in a role, you cannot include SYS_MANAGE_ROLES_ROLE in the administrators list when creating a role with role administrators granted membership in the role (WITH ADMIN option). It can, however, be included when creating a role with role administrators not granted membership in the role (WITH ADMIN ONLY option).

Example

This statement creates the role `Sales` with no role administrators specified. Any user with the MANAGE ROLES system privilege is a default administrator of this role.

```
CREATE ROLE Sales
```

This statement creates the role `Marketing` with `<Jane>` and `<Bob>` acting as role administrators, but not granted membership in the role. It also allows global role administrators to manage the role.

```
CREATE ROLE Marketing WITH ADMIN ONLY SYS_MANAGE_ROLES_ROLE, Jane, Bob
```

Related Information

Role and Global Role Administrators [page 19]
CREATE ROLE Statement [page 307]

1.2.1.2 Converting an Existing User to a User-Extended Role

You can extend an existing user ID to act as a role. This is useful when you have a user who is assigned a set of system and object-level privileges that you want to grant to another user.

Prerequisites

MANAGE ROLES system privilege.

Context

If an existing ID has login privileges, the user-extended role retains the login privileges.

When converting a user to act as a role, you can appoint administrators for the role, and indicate whether they are also to be members of the role. If you do not specify any administrators, the global role administrator (any user granted the MANAGE ROLES system privilege) becomes the default administrator of the role.

However, if at least one role administrator is specified, global role administrators cannot manage the role because the SYS_MANAGE_ROLES_ROLE system privilege is not automatically granted to the role with administrative rights. For this reason, SAP strongly recommends that you either do not define any role administrators when creating or converting a role (add them after creation), or explicitly grant the SYS_MANAGE_ROLES_ROLE system privilege with administrative rights only along with any role administrators during the process.

You can add or remove role administrators can be added and removed after converting a user. If you attempt to convert a user using a user ID that does not exist, the statement fails.

Procedure

To convert an existing user, execute one of these statements:

Convert Condition	Statement
Global role administrator only; **no role administrators**	`CREATE ROLE FOR USER <userID>`

Role administrators with no role membership;
no global role administrator
	`CREATE ROLE FOR USER <userID>`
	`WITH ADMIN ONLY <admin_name [,...]>`

Role administrators with role membership;	`CREATE ROLE FOR USER <userID>`

Convert Condition	Statement
no global role administrator*	`WITH ADMIN <admin_name [,...]>`

Role administrators with no role membership;
global role administrator*
	`CREATE ROLE FOR USER <userID>`
	`WITH ADMIN ONLY SYS_MANAGE_ROLES_ROLE, <admin_name [,...]>`

*Since global role administrators cannot be granted membership in a role, you cannot include SYS_MANAGE_ROLES_ROLE in the administrators list when creating a role with role administrators granted membership in the role (WITH ADMIN option). It can, however, be included when creating a role with role administrators not granted membership in the role (WITH ADMIN ONLY option).

Example

This statement extends user `Sales1` to act as a role. Since no role administrators are specified, any user with the MANAGE ROLES system privilege can administrator the role.

```
CREATE ROLE FOR USER Sales1
```

This statement extends the user `Marketing1` to act as a role, with `<Jane>` and `<Bob>` acting as role administrators. It also allows global role administrators to manage the role.

```
CREATE ROLE FOR USER Marketing1 WITH ADMIN ONLY SYS_MANAGE_ROLES_ROLE, Jane, Bob
```

Related Information

CREATE ROLE Statement [page 307]

1.2.1.3 Converting a User-Extended Role Back to a User

You can convert a user-extended role back to a regular user.

Prerequisites

Administrative rights over the user-extended role being converted.

Context

The user retains any login privileges, system privileges, and roles that are granted to the user-extended role. The user remains as the owner of the objects that were created after the user was extended to act as a role. Any members of the user-extended role are immediately revoked.

A minimum number of role or global role administrators (as defined by the `MIN_ROLE_ADMINS` database option) with a login password must exist for each role at all times. When converting a user-extended role back to a user, all dependent roles of the user-extended role must continue to meet this minimum requirement, or the conversion fails.

Procedure

To convert a user-extended role back to a user, execute one of these:

Convert Condition	Statement
Role has not been granted any members.	`DROP ROLE FROM USER <role_name>`

Role has been granted members.	`DROP ROLE FROM USER <role_name>`
	`WITH REVOKE`

1.2.1.4 Adding a User-Defined Role to a User or Role

Add membership in a user-defined role to a user or role (grantee), with or without administrative rights.

Prerequisites

Administrative privilege over the role being granted.

Context

A user-defined role can be granted with or without administrative rights. When granted with administrative rights (that is using the WITH ADMIN OPTION clause), a user can manage (grant, revoke, and drop) the role, as well as use any of the underlying system and object-level privileges of the role. When granted with administrative rights only (using the WITH ADMIN ONLY OPTION clause), a user can manage the role, but cannot use its underlying system and object-level privileges. When granted without any administrative rights, a user can use its underlying system and object-level privileges, but cannot manage the role.

When a user is granted membership in a role, the user inherits all underlying system privileges and roles of the role, including any object-level permissions on tables, views, and procedures.

When a role is granted to another role, all members of the role being granted (the child role) automatically become members of the receiving role (parent role), and inherit all underlying system privileges and roles of the parent role, including those on tables, views, and procedures. Existing members of the parent role do not become members of the child role or inherit any of its underlying system privileges and roles.

Procedure

To grant a user-defined role to a grantee, execute one of these statements:

Grant Type	Statement

Membership in the role
along with full administrative rights
to the role
	`GRANT ROLE <role_name> TO <grantee [,...]>`
	`WITH ADMIN OPTION`

Administrative rights
to the role only
	`GRANT ROLE <role_name> TO <grantee [,...]>`
	`WITH ADMIN ONLY OPTION`

Membership in the role,
but with no administrative rights
to the role
	`GRANT ROLE <role_name> TO <grantee [,...]>`
	`WITH NO ADMIN OPTION`

Example

- There are three users: User1, User2, User3.
- There are four roles: Role1, Role2, Role3, Role4.
- There are two system privileges: Priv1, Priv2.
- Role1 is granted Priv1 and Role3.
- User2 and User3 are members of Role1.
- Role2 is granted Priv2 and Role4.
- User3 is a member of Role2.

You execute the following statement:

```
GRANT ROLE Role1 TO User1 WITH ADMIN OPTION
```

User1 becomes a member of Role1.

As a member of Role1, User1 inherits Priv1 and (indirectly) all system privileges and roles from Role3.

User1 can also administer Role1.

You execute the following statement:

```
GRANT ROLE Role2 TO Role1 WITH ADMIN OPTION
```

Role1 becomes a member of Role2.

As members of Role1, User2, User3, and User1 (from previous grant) inherit the following from Role2: Priv2 and (indirectly) all system privileges and roles of Role4.

As a member of Role2, User3 does not become a member of Role1 and does not inherit any system privileges or roles of Role1.

User1, User2, and User3 can administer Role2.

Related Information

GRANT ROLE Statement [page 323]

1.2.1.5 Removing Members from a User-Defined Role

Remove a user or role as a member of a role. The user or role loses the ability to use any underlying system privileges or roles of a role, along with the ability to administer the role, if granted.

Prerequisites

Administrative privilege over the role being managed.

Context

A minimum number of role or global role administrators (as defined by the `MIN_ROLE_ADMINS` database option) with a login password must exist for each role at all times. If the member is an administrator of the role and his or her removal violates the minimum requirement, the removal fails.

Procedure

To remove membership in a user-defined role from a grantee, execute one of these statements:

Revoke Type	Statement

Role membership and
all administrative rights to the role
	`REVOKE ROLE <role_name>`
	`FROM <grantee [,...]>`

Administrative rights
to the role only
	`REVOKE ADMIN OPTION FOR ROLE <role_name>`
	`FROM <grantee [,...]>`

Related Information

REVOKE ROLE Statement [page 340]

1.2.1.6 Deleting a User-Defined Role

Delete a user-defined role from the database as long as all dependent roles retain the minimum required number of administrator users with active passwords. If the minimum value is not maintained, the command fails.

Prerequisites

- Administrative privilege over the role being dropped.
- If the role being dropped is a user-defined role, the role does not own any objects.

Context

If a user-extended role is converted back to a user, the objects owned are not deleted; the converted user continues to own them.

The type of role being deleted and whether it was granted to users determines the clauses required by the DROP statement.

FROM USER required when deleting a user-extended role.
WITH REVOKE required to delete a role that has been granted to multiple users and roles.

Procedure

To delete a user-defined role, execute one of these statements:

Delete Condition	Statement
User-defined role has not been granted any members.	DROP ROLE <role_name>
User-extended role has been granted members.	DROP ROLE <role_name> WITH REVOKE
User-extended role has not been granted any members*.	DROP ROLE FROM USER <role_name>
User-extended role has been granted members*.	DROP ROLE FROM USER <role_name> WITH REVOKE

*User-extended role becomes a regular user.

Related Information

DROP ROLE Statement [page 313]

1.2.1.7 Role and Global Role Administrators

Role administrators and global role administrators grant and revoke user-defined roles to users and other roles. You can add and remove role and global role administrators on a role as needed.

There is no maximum number of role administrators that can be granted to a single role. However, there is a minimum number, as specified by the configurable `MIN_ROLE_ADMINS` database option. This minimum requirement is validated before you can revoke a role administrator or global role administrator from a role. The minimum number of role administrators can be set to any value between 1 (default) and 10.

A role administrator can be a user, a user-extended role, or a user-defined role.

Global role administrators include users who are granted the MANAGE ROLES system privilege. Global role administrators can administer any role to which the SYS_MANAGE_ROLES_ROLE system privilege has been granted with administrative rights.

Both role and global role administrators can grant, revoke, and drop roles, and can add or remove role and global role administrators to and from a role. A role administrator can be a user or a role and does not require the MANAGE ROLES system privilege to administer a role.

You can appoint role administrators either when creating the role or after the role has been created, and indicate whether they are also to be members of the role. If you do not specify any administrators, the global role administrator is, by default, the administrator of the role.

However, if at least one role administrator is specified, global role administrators cannot manage the role because the SYS_MANAGE_ROLES_ROLE system privilege is not automatically granted to the role with administrative rights. For this reason, SAP strongly recommends that you either do not define any role administrators when creating or converting a role (add them after creation), or explicitly grant the SYS_MANAGE_ROLES_ROLE system privilege with administrative rights only along with any role administrators during the process.

If you do not specify a role administrator when you create a role, the global role administrator (SYS_MANAGE_ROLES_ROLE system privilege) is automatically granted to the role with administrative-only rights.

If you later add role administrators to a role originally created with no role administrators, the global role administrator (SYS_MANAGE_ROLES_ROLE system privilege) may or may not be removed, depending on how you add the role administrators. If you use the `GRANT` statement, the SYS_MANAGE_ROLES_ROLE system privilege remains granted to the role. However, if you use the `CREATE OR REPLACE` statement, the SYS_MANAGE_ROLES_ROLE system privilege is removed if it is not explicitly included in the new list of role administrators.

> i Note
>
> You cannot remove the SYS_MANAGE_ROLES_ROLE system privilege from a role if so doing results in a failure to meet the minimum number of role administrators defined.

By default, the SYS_MANAGE_ROLES_ROLE system privilege is not granted to compatibility roles (SYS_AUTH_*_ROLE). Therefore, to allow global role administrators to manage a compatibility role, you must explicitly grant SYS_MANAGE_ROLES_ROLE with administrative rights only to the role.

In this section:

Remove the global role administrator from a role.

Minimum Number of Role Administrators [page 31]
The `MIN_ROLE_ADMINS` database option is a configurable value that ensures you never create a scenario where there are no users and roles left with sufficient system privilege to manage the remaining users and roles.

DBA User Unable to Administer a Role [page 33]
Under several circumstances, the DBA user might be unable to manage (grant, revoke, or drop) a role.

1.2.1.7.1 Adding a Role Administrator When Creating a Role

Specify a role administrator when creating a new role.

Prerequisites

MANAGE ROLES system privilege.

Context

If you specify at least one role administrator when you create a role, global role administrators cannot manage the role unless explicitly specified.

For this reason, SAP strongly recommends that you consider always adding the global role administrator to the list of role administrators.

Procedure

To add role administrators during the creation process, execute one of these statements:

Create Type	Statement

Administrative rights only;
no role membership
	`CREATE ROLE <role_name>`
	`WITH ADMIN ONLY <admin_name [,...]>`

Role and global role administrators granted administrative rights only; no role membership*	`CREATE ROLE <role_name>`
	`WITH ADMIN ONLY SYS_MANAGE_ROLES_ROLE, <admin_name [,...]>`

Create Type	Statement

Administrative rights along with role membership	`CREATE ROLE <role_name>`
	`WITH ADMIN <admin_name [,...]>`

*Since global role administrators cannot be granted membership in a role, you cannot include SYS_MANAGE_ROLES_ROLE in the administrators list when you create a role with role administrators granted membership in the role (WITH ADMIN option).

Example

Execute this statement to make `Joe` and `Bob` role administrators of the `Sales` role:

```
CREATE ROLE Sales WITH ADMIN Joe, Bob
```

Because it uses the WITH ADMIN clause, both `Joe` and `Bob` can both grant and revoke the role, as well as use the underlying system privileges of the role. If the WITH ADMIN ONLY clause were used, both `Joe` and `Bob` would be able to only grant and revoke the role.

Execute this statement to make `Joe` and `Bob` role administrators of the `Sales` role, as well as to allow global role administrators to manage the role:

```
CREATE ROLE Sales WITH ADMIN ONLY SYS_MANAGE_ROLES_ROLE, Joe, Bob
```

Related Information

CREATE ROLE Statement [page 307]

1.2.1.7.2 Adding the Global Role Administrator When Creating a Role

Allow global role administrators to administer a new role.

Prerequisites

MANAGE ROLES system privilege.

Context

If you specify at least one role administrator when you create a role, global role administrators cannot manage the role unless explicitly specified.

For this reason, SAP strongly recommends that you consider always adding the global role administrator to the list of role administrators.

Procedure

To add the global role administrator during the creation process, execute one of these statements:

Create Type	Statement
Global role administrator only; **no role administrators**	`CREATE ROLE <role_name>`

Both role and global role administrators*	`CREATE ROLE <role_name>`
	`WITH ADMIN ONLY SYS_MANAGE_ROLES_ROLE, <admin_name [,...]>`

*Global role administrator can have only administrative rights (WITH ADMIN ONLY) on a role. Therefore, if you specify both role and global role administrators when creating a role, only the WITH ADMIN ONLY clause is valid.

Example

Execute this statement to create the `Sales` role and allow only global role administrators to manage it:

```
CREATE ROLE Sales
```

Execute this statement to make `Joe` and `Bob` role administrators of the `Sales` role, with administrative rights only, as well as to allow global role administrators to manage the role:

```
CREATE ROLE Sales WITH ADMIN ONLY SYS_MANAGE_ROLES_ROLE, Joe, Bob
```

1.2.1.7.3 Adding Role Administrators to an Existing Role

Add role administrators to an existing role. There is no maximum number of role administrators that can be granted to a single role.

Prerequisites

Administrative privilege over the role, or the MANAGE ROLES system privilege, if the role has a global role administrator.

Procedure

To add role administrators, execute one of these statements:

Grant Type	Statement

Administrative privileges only	`GRANT ROLE <role_name> TO <admin_name [,...]>`
	`WITH ADMIN ONLY OPTION`

Administrative privileges and role membership	`GRANT ROLE <role_name> TO <admin_name [,...]>`
	`WITH ADMIN OPTION`

Example

Execute this statement to make `Mary` and `Bob` role administrators of the `Sales` role.

```
GRANT ROLE Sales TO Mary, Bob WITH ADMIN ONLY OPTION
```

Each can administer the role, but not use its underlying system privileges because of the WITH ADMIN ONLY OPTION clause.

Execute this statement to make `Sarah` a role administrator of the `Sales` role with the ability to both administer the role and use its underlying system privileges because of the WITH ADMIN OPTION clause.

```
GRANT ROLE Sales TO Sarah WITH ADMIN OPTION
```

Related Information

GRANT ROLE Statement [page 323]

1.2.1.7.4 Adding the Global Role Administrator to an Existing Role

Add the global role administrator to an existing role.

Prerequisites

Administrative privilege over the role.

Context

You can grant the global role administrator to a role with administrative rights only (WITH ADMIN ONLY option).

Procedure

To reinstate the global role administrator on a role, execute:

```
GRANT ROLE <role_name> TO SYS_MANAGE_ROLES_ROLE
WITH ADMIN ONLY OPTION
```

Related Information

GRANT ROLE Statement [page 323]

1.2.1.7.5 Making a User or Role a Global Role Administrator

Allow a user or role to act as a global role administrator.

Prerequisites

MANAGE ROLES system privilege granted with administrative rights.

Context

To become a global role administrator, you must be granted the MANAGE ROLES system privilege. Administrative rights on the MANAGE ROLES system privilege are not required to act as a global role administrator. If the MANAGE ROLES system privilege is granted to a role, all members of the role inherit the system privilege, and thus the ability to act as a global role administrator.

Procedure

To grant the MANAGE ROLES system privilege execute:

```
GRANT MANAGE ROLES TO <grantee [,...]>
```

Related Information

GRANT System Privilege Statement [page 329]

1.2.1.7.6 Replacing Existing Role Administrators on a Role

Replace current role administrators with new administrators.

Prerequisites

Administrative privilege over the role, or the MANAGE ROLES system privilege, if the role has a global role administrator.

Context

Replacing role administrators involves changing the users and roles who can act as administrators, and their level of administrative rights on the role. Depending on the extent of the replacement, there are two approaches you can take. Each approach has different net effects on role and global administrators. The first approach allows you to selectively replace the administrators of an existing role. The second approach allows you to completely replace all existing role administrators. Using the second approach includes replacing the global role administrator.

The first approach is a two-step process: Add new role administrators, then remove existing administrators from the role. You must meet the minimum number of administrators requirement throughout; therefore, SAP recommends that you add new administrators before you remove existing ones. If the role has a global role administrator, it is retained unless you explicitly remove it.

The second approach is a one-step process, but has a much broader impact: Define a new list of role administrators. All current role administrators are overwritten with new role administrators. If any current role administrators are to continue in this capacity, you must include them in the list of replacement role administrators. The list replaces all existing administrators, with the following behavior:

- All existing role administrators granted the WITH ADMIN OPTION that are not included on the new role administrators list become members of the role with no administrative rights.
- All existing role administrators granted the WITH ADMIN ONLY OPTION that are not included on the new role administrators list are removed as members of the role.
- An existing role administrator included on the new role administrators list retains his or her original administrative rights if they are higher than the replacement rights. For example, the new role administrators are granted WITH ADMIN ONLY rights. `User1`, who was originally granted the role with WITH ADMIN rights, and is included on the new list, retains the higher WITH ADMIN rights.
- If the role has a global role administrator, it is removed from the role unless you explicitly include it on the new role administrators list.
- If new role administrators are granted WITH ADMIN rights, an existing global role administrator cannot be included in the list, since it cannot be granted WITH ADMIN rights. It is removed from the role.

You can issue the replacement role command as long as the replacement administrative option is equal to or higher than the current level. To lower the administrative level, first remove (revoke) all role administrators from the role, and then regranted them.

A minimum number of role or global role administrators (as defined by the `MIN_ROLE_ADMINS` database option) with a login password must exist for each role at all times. When replacing role administrators, if the number of replacement administrators violates the minimum requirement, the replacement fails.

Procedure

To replace role administrators, execute one of:

Replacement Option	Statement
Replace select role administrators **with administrative only rights;** **no role membership**	○ `GRANT ROLE <role_name> TO <admin_name [,...]> WITH ADMIN ONLY OPTION` ○ `REVOKE ADMIN OPTION FOR ROLE <role_name> FROM <admin_name [,...]>`
Replace select role administrators **with administrative and role membership**	○ `GRANT ROLE <role_name> TO <admin_name [,...]> WITH ADMIN OPTION` ○ `REVOKE ADMIN OPTION FOR ROLE <role_name> FROM <admin_name [,...]>`

Replace all role administrators
with administrative rights only; no role membership.
	`CREATE OR REPLACE ROLE <role_name>`
	`WITH ADMIN ONLY <admin_name [,...]>`

Replacement Option	Statement
Remove the global role administrator, if exists.	

Replace all role administrators with
administrative rights and role membership.
Remove the global role administrator, if exists.
	`CREATE OR REPLACE ROLE <role_name>`
	`WITH ADMIN <admin_name [,...]>`

Replace all role administrators with
administrative rights only
including the global role administrator.*
	`CREATE OR REPLACE ROLE <role_name>`
	`WITH ADMIN ONLY SYS_MANAGE_ROLES_ROLE, <admin_name [,...]>`

Replace all role administrators **with full administrative rights.** **Restore the global role administrator to the role***	○ `CREATE OR REPLACE ROLE <role_name> WITH ADMIN <admin_name [,...]>` ○ `GRANT ROLE <role_name> TO SYS_MANAGE_ROLES_ROLE WITH ADMIN ONLY OPTION`

*SYS_MANAGE_ROLES_ROLE can be granted to a role only using the WITH ADMIN ONLY option. Therefore, when the CREATE OR REPLACE statement includes the WITH ADMIN ONLY option, SYS_MANAGE_ROLES_ROLE can be included in the administrator list. When the CREATE OR REPLACE statement uses the WITH ADMIN option, you must issue a separate grant statement to grant SYS_MANAGE_ROLES_ROLE to the role using the WITH ADMIN ONLY option.

Example

`Sales` has `Mary` and `Bob` as role administrators with full administrative rights. `Sales` has a global role administrator.

Execute these statements to remove `Bob` as a role administrator and replace him with `Sarah` and `Jeff`, with the same administrative rights. `Bob` remains a member of `Sales` with no administrative rights.

```
GRANT ROLE sales TO Sarah, Jeff WITH ADMIN OPTION
REVOKE ADMIN OPTION FOR ROLE Sales FROM Bob
```

Execute these statements to replace the existing role administrators (`Mary` and `Bob`) with `Sarah` and `Jeff`, with full administrative rights. Since the global role administrator cannot be included on the list (cannot be granted with full administrative rights), it must be explicitly regranted to the role after replacing the role administrators.

```
CREATE OR REPLACE ROLE Sales WITH ADMIN Sarah, Jeff
GRANT ROLE sales TO SYS_MANAGE_ROLES_ROLE WITH ADMIN ONLY OPTION
```

Execute these statements to replace the existing role administrators (`Mary` and `Bob`) with `Bob` and `Sarah` with administrative rights only. To preserve the global role administrator, it must be included on the list. Since Bob is to remain as a role administrator, and originally had higher administrative rights than the new role administrators, he retains the original higher administrative rights.

```
CREATE OR REPLACE ROLE Sales WITH ADMIN ONLY Bob, Sarah, SYS_MANAGE_ROLES_ROLE
```

Related Information

GRANT ROLE Statement [page 323]
REVOKE ROLE Statement [page 340]
CREATE ROLE Statement [page 307]

1.2.1.7.7 Removing a Role Administrator from a Role

Remove a role administrator from a role.

Prerequisites

Administrative privilege over the role.

Context

A minimum number of role or global role administrators (as defined by the `MIN_ROLE_ADMINS` database option) with a login password must exist for each role at all times. You can remove role administrators only as long as the this minimum is still met after removal.

When removing a role administrator, if role administration was originally granted to the user using the WITH ADMIN OPTION clause, revoking role administration removes only their ability to manage the role (grant, revoke, drop), not the ability to use the underlying system privileges of the role (membership). However, if role administration was originally granted to the user using the WITH ADMIN ONLY OPTION clause, revoking role administration has the same effect as revoking the role entirely, as there was no membership associated with the role.

Procedure

To remove a role administrator from a role, execute one of these statements:

Removal Type	Statement

Remove role administrator,
but retain membership in the role.
	`REVOKE ADMIN OPTION FOR ROLE <role_name>`
	`FROM <admin_name [,...]>`

Remove role administrator
along with membership in the role.
	`REVOKE ROLE <role_name>`
	`FROM <admin_name [,...]>`

Example

This example assumes that both `Mary` and `Sarah` are currently role administrators of the `Sales` role. `Mary` has been granted both membership in the role and the ability to administer the role. `Sarah`, however, has been granted only the ability to administer the role, not membership. Due to the different administration levels granted, executing this statement to revoke administrative rights from the `Sales` role has a different impact on each administrator:

```
REVOKE ADMIN OPTION FOR ROLE Sales FROM Mary, Sarah
```

It results in the loss of `Mary's` ability to administer the `Sales` role, but retains her membership of the role. It completely removes the `Sales` role from `Sarah`.

Related Information

REVOKE ROLE Statement [page 340]

1.2.1.7.8 Removing the Global Role Administrator from a Role

Remove the global role administrator from a role.

Prerequisites

Administrative privilege over the role.

Context

A minimum number of role or global role administrators (as defined by the `MIN_ROLE_ADMINS` database option) with a login password must exist for each role at all times. You can remove the global role administrator from a role as long as this minimum is still met for the role.

Procedure

To remove the global role administrator from a role, execute:

```
REVOKE ROLE <role_name>
FROM SYS_MANAGE_ROLES_ROLE
```

Related Information

REVOKE ROLE Statement [page 340]

1.2.1.7.9 Minimum Number of Role Administrators

The `MIN_ROLE_ADMINS` database option is a configurable value that ensures you never create a scenario where there are no users and roles left with sufficient system privilege to manage the remaining users and roles.

This value applies to the minimum number of role administrators for each role, not for the total number of roles, and is considered when you:

- Create or Revoke roles
- Drop users or roles
- Change a user's password to null

> i Note
>
> Users or roles without passwords cannot be administrators.

When you attempt to change this value, the system validates that each existing role continues to have at least as many role administrators as defined by the new value. If even one role fails to meet this requirement, the statement fails. Similarly, when dropping users, if the number of remaining administrators drops below the designated minimum value, the statement fails.

> i Note
>
> Locked accounts are not considered when counting the number of administrators for a role.

Example 1

`MIN_ROLE_ADMINS` value is 2

`Role1` has two administrators and `Role2` has three administrators.

If you reduce the value to 1, the command succeeds because both roles still have the new designated minimum number of role administrators. However, if you increase the value to 3, the command fails because `Role1` no longer has sufficient administrators to meet the new minimum value.

Example 2

`MIN_ROLE_ADMINS` value is 4

`Role1` has six administrators and `Role2` has four administrators.

If you drop a user from `Role1`, the command succeeds because `Role1` still has sufficient administrators to meet the minimum value. However, if you drop a user from `Role2`, the command fails because `Role2` no longer has sufficient administrators to meet the minimum value.

In this section:

Related Information

Automatic Unlocking of User Accounts [page 156]
MIN_ROLE_ADMINS Option [page 357]

1.2.1.7.9.1 Setting the Minimum Number of Role Administrators

Set the minimum number of role administrators required to manage each role.

Prerequisites

SET ANY SECURITY OPTION system privilege.

Context

The minimum number of role administrators is a configurable database option that you can set to any integer between 1 (the default) and 10. You cannot change this value if so doing results in the number of role administrators for any single role not meeting the new minimum value. You also cannot temporarily set this option.

This value applies to each role, not all roles in total. For example, if there are 20 roles and the minimum number of role administrators is set to 2, each of the 20 roles must have a minimum of 2 role administrators defined, not 2 role administrators defined to administer the 20 roles in total.

Procedure

To change the minimum number of role administrators, execute:

```
SET OPTION Public.min_role_admins = <value>
```

Related Information

Automatic Unlocking of User Accounts [page 156]
MIN_ROLE_ADMINS Option [page 357]

1.2.1.7.10 DBA User Unable to Administer a Role

Under several circumstances, the DBA user might be unable to manage (grant, revoke, or drop) a role.

This situation occurs when:

- The global role administrator has been removed from the role; or
- The DBA user is not defined as a role administrator for the role.

To resolve the issue, grant the global role administrator to the role (recommended) or add the DBA user as a role administrator for the role.

Related Information

GRANT ROLE Statement [page 323]
Adding Role Administrators to an Existing Role [page 24]
Adding the Global Role Administrator to an Existing Role [page 25]

1.2.2 System Roles

System roles are built-in roles that are automatically created in each new database.

System roles:

- Cannot be dropped.
- Cannot have their default underlying system privileges modified or revoked.
- Can have additional roles and system privileges granted to (or revoked from).
- Cannot be granted with administrative rights (WITH ADMIN OPTION or WITH ADMIN ONLY OPTION clauses).
- Have no a password assigned, so users cannot connect to the database as a grantable system role.
- Do not own objects, except for the SYS, dbo, and rs_systabgroup role.

In this section:

Granting the dbo System Role [page 35]
The dbo system role owns many system stored procedures and views.

Granting the diagnostics System Role [page 36]
Members of the diagnostics system role inherit SELECT, INSERT, UPDATE, DELETE, and ALTER privileges on diagnostic tables and views.

Granting the PUBLIC System Role [page 36]
The PUBLIC system role has SELECT privilege on a set of system tables and EXECUTE privilege on system procedures.

Granting the rs_systabgroup System Role [page 37]
The rs_systabgroup system role owns tables and system procedures that are required for Replication Server, and grants users the underlying system privileges to perform Replication Server functionality.

Granting the SYS System Role [page 38]
The SYS system role owns the system tables and views for the database, which contain the full description of database schema, including all database objects and user IDs.

Granting the SYS_REPLICATION_ADMIN_ROLE [page 39]
The SYS_RUN_REPLICATION_ADMIN_ROLE system role is required for performing administration tasks that are related to replication, such as granting replication roles, managing publications, subscriptions, synchronization users and profiles, managing message types, setting replication-related options, and so on.

Granting the SYS_RUN_REPLICATION_ROLE [page 40]
The SYS_RUN_REPLICATION_ROLE system role is required for performing replication tasks using `dbremote`, and synchronization tasks using `dbmlsync`. The SYS_RUN_REPLICATION_ROLE system role is active only for users who connect through these utilities.

Granting the SYS_SPATIAL_ADMIN_ROLE System Role [page 42]
The SYS_SPATIAL_ADMIN_ROLE system role grants users the ability to create, alter, drop, or comment on spatial reference systems and spatial units of measure. SYS_SPATIAL_ADMIN_ROLE is the owner of all spatial objects.

Revoking a System Role [page 43]
Revokes a system role from a user or role.

1.2.2.1 Granting the dbo System Role

The dbo system role owns many system stored procedures and views.

Prerequisites

MANAGE ROLES system privilege.

Context

By default, the dbo system role is a member of the SYS system role and SYS_AUTH_RESOURCE_ROLE compatibility role with no administrative rights. It is also a member of the SYS_AUTH_DBA_ROLE compatibility role with full administrative rights.

You can grant the dbo system role to other roles only with no administrative rights (WITH NO ADMIN OPTION clause). The WITH ADMIN OPTION and WITH ADMIN ONLY OPTION clauses are invalid for the dbo system role.

You can grant system privileges and roles to, and revoke them from, the dbo system role, including the default roles.

Procedure

To grant the dbo system role, execute:

```
GRANT ROLE dbo TO <grantee [,...]>
```

Related Information

GRANT ROLE Statement [page 323]

1.2.2.2 Granting the diagnostics System Role

Members of the diagnostics system role inherit SELECT, INSERT, UPDATE, DELETE, and ALTER privileges on diagnostic tables and views.

Prerequisites

MANAGE ROLES system privilege.

Context

You can grant the diagnostics system role to other roles only with no administrative rights (WITH NO ADMIN OPTION clause). The WITH ADMIN OPTION and WITH ADMIN ONLY OPTION clauses are invalid for the diagnostics system role.

You can grant system privileges and roles to, and revoke them from, the diagnostics system role.

Procedure

To grant the diagnostics system role, execute:

```
GRANT ROLE diagnostics TO <grantee [,...]>
```

Related Information

GRANT ROLE Statement [page 323]

1.2.2.3 Granting the PUBLIC System Role

The PUBLIC system role has SELECT privilege on a set of system tables and EXECUTE privilege on system procedures.

Prerequisites

MANAGE ROLES system privilege.

Context

By default, the PUBLIC system role is a member of the dbo and SYS system roles, with no administrative rights. As a member of the SYS role, it has read access for some system tables and views, so any user of the database can see information about the database schema. To restrict this access, revoke PUBLIC's membership in the SYS system role.

Any new user ID is automatically a member of the PUBLIC system role and inherits any privileges that are specifically granted to that role. Although you can remove a user from the PUBLIC system role, SAP recommends that you do not, as doing so might impact a user's ability to run system stored procedures.

You can grant the PUBLIC system role to other roles only with no administrative rights (WITH NO ADMIN OPTION clause). The WITH ADMIN OPTION and WITH ADMIN ONLY OPTION clauses are invalid for the PUBLIC system role.

You can grant system privileges and roles to, and revoke them from, the PUBLIC system role, including the default roles.

Procedure

To grant the PUBLIC system role, execute:

```
GRANT ROLE PUBLIC TO <grantee [,...]>
```

Related Information

GRANT ROLE Statement [page 323]

1.2.2.4 Granting the rs_systabgroup System Role

The rs_systabgroup system role owns tables and system procedures that are required for Replication Server, and grants users the underlying system privileges to perform Replication Server functionality.

Prerequisites

MANAGE ROLES system privilege.

Context

You can grant the rs_systabgroup system role to other roles only with no administrative rights (WITH NO ADMIN OPTION clause). The WITH ADMIN OPTION and WITH ADMIN ONLY OPTION clauses are invalid for the rs_systabgroup system role.

You can grant system privileges and roles to, and revoke them from, the rs_systabgroup system role.

Procedure

To grant the rs_systabgroup system role, execute:

```
GRANT ROLE rs_systabgroup TO <grantee [,...]>
```

Related Information

GRANT ROLE Statement [page 323]

1.2.2.5 Granting the SYS System Role

The SYS system role owns the system tables and views for the database, which contain the full description of database schema, including all database objects and user IDs.

Prerequisites

MANAGE ROLES system privilege.

Context

By default, the SYS system role is granted the dbo and PUBLIC system roles with no administrative rights. However, members of the dbo and PUBLIC system roles do not inherit any system privileges directly or indirectly granted to the SYS system role.

You can grant the SYS system role to other roles only with no administrative rights (WITH NO ADMIN OPTION clause). The WITH ADMIN OPTION and WITH ADMIN ONLY OPTION clauses are invalid for the SYS system role.

You cannot grant additional system privileges to, or revoke them from, the SYS system role.

Procedure

To grant the SYS system role, execute:

```
GRANT ROLE SYS TO <grantee [,...]>
```

Related Information

GRANT ROLE Statement [page 323]

1.2.2.6 Granting the SYS_REPLICATION_ADMIN_ROLE

The SYS_RUN_REPLICATION_ADMIN_ROLE system role is required for performing administration tasks that are related to replication, such as granting replication roles, managing publications, subscriptions, synchronization users and profiles, managing message types, setting replication-related options, and so on.

Prerequisites

MANAGE ROLES system privilege.

Context

By default, the SYS_REPLICATION_ADMIN_ROLE system role is granted these system privileges with no administrative rights:

- CREATE ANY PROCEDURE
- CREATE ANY TABLE
- DROP ANY TABLE
- DROP ANY PROCEDURE
- MANAGE ANY OBJECT PRIVILEGE
- MANAGE ANY USER
- MANAGE ANY WEB SERVICE
- MANAGE REPLICATION
- MANAGE ROLES
- SERVER OPERATOR
- SELECT ANY TABLE
- SET ANY SYSTEM OPTION
- SET ANY PUBLIC OPTION

- SET ANY USER DEFINED OPTION

You cannot revoke this default set of system privileges from the SYS_RUN_REPLICATION_ADMIN_ROLE system role, but you can grant additional system privileges and roles to, and revoke them from, the SYS_RUN_REPLICATION_ADMIN_ROLE system role.

You can grant the SYS_RUN_REPLICATION_ADMIN_ROLE system role to other roles only with no administrative rights (WITH NO ADMIN OPTION clause). The WITH ADMIN OPTION and WITH ADMIN ONLY OPTION clauses are invalid for the SYS_RUN_REPLICATION_ADMIN_ROLE system role.

Procedure

To grant the SYS_REPLICATION_ADMIN_ROLE system role, execute:

```
GRANT ROLE SYS_REPLICATION_ADMIN_ROLE TO <grantee [,...]>
```

Related Information

GRANT ROLE Statement [page 323]

1.2.2.7 Granting the SYS_RUN_REPLICATION_ROLE

The SYS_RUN_REPLICATION_ROLE system role is required for performing replication tasks using `dbremote`, and synchronization tasks using `dbmlsync`. The SYS_RUN_REPLICATION_ROLE system role is active only for users who connect through these utilities.

Prerequisites

MANAGE REPLICATION system privilege.

Context

The SYS_RUN_REPLICATION_ROLE system role is a member of the SYS_AUTH_DBA_ROLE compatibility role with full administrative rights.

It is also granted these system privileges with no administrative rights:

- SELECT ANY TABLE
- SET ANY USER DEFINED OPTION

- SET ANY SYSTEM OPTION
- BACKUP DATABASE
- MONITOR

You cannot revoke this default set of system privileges from the SYS_RUN_REPLICATION_ROLE system role, but you can grant additional system privileges and roles to, and revoke them from, the SYS_RUN_REPLICATION_ROLE system role.

By default, the SYS_AUTH_DBA_ROLE compatibility role is granted to the SYS_RUN_REPLICATION_ROLE system role to address any possible requirements for additional system privileges to perform other replication related authorized tasks over and above the above-noted explicitly granted system privileges. However, SAP recommends that you revoke the SYS_AUTH_DBA_ROLE compatibility role from SYS_RUN_REPLICATION_ROLE system role and explicitly grant those specific additional system privileges or roles identified for other replication tasks to the SYS_RUN_REPLICATION_ROLE system role.

You can grant the SYS_RUN_REPLICATION_ROLE system role to other roles only with no administrative rights (WITH NO ADMIN OPTION clause). The WITH ADMIN OPTION and WITH ADMIN ONLY OPTION clauses are invalid for the SYS_RUN_REPLICATION_ROLE system role.

By default, when granting SYS_RUN_REPLICATION_ROLE, underlying system privileges are inherited by members of the receiving group. To prevent inheritance, include the WITH NO SYSTEM PRIVILEGE INHERITANCE clause for this system role only.

The `MIN_ROLE_ADMINS` database option ensures that a designated number of users who can grant the MANAGE REPLICATION system privilege to, and revoke from, other users always exists in the database.

Procedure

To grant the SYS_RUN_REPLICATION_ROLE system role, execute one of these statements:

Inheritance Type	Statement
With inheritance	`GRANT ROLE SYS_RUN_REPLICATION_ROLE TO <grantee [,...]>`

With no inheritance	`GRANT ROLE SYS_RUN_REPLICATION_ROLE TO <grantee [,...]>`
	`WITH NO SYSTEM PRIVILEGE INHERITANCE`

Related Information

GRANT ROLE Statement [page 323]

1.2.2.8 Granting the SYS_SPATIAL_ADMIN_ROLE System Role

The SYS_SPATIAL_ADMIN_ROLE system role grants users the ability to create, alter, drop, or comment on spatial reference systems and spatial units of measure. SYS_SPATIAL_ADMIN_ROLE is the owner of all spatial objects.

Prerequisites

MANAGE ROLES system privilege.

Context

By default, the SYS_SPATIAL_ADMIN_ROLE system role is granted the MANAGE ANY SPATIAL OBJECT system privilege with no administrative rights.

You can grant the SYS_SPATIAL_ADMIN_ROLE system role to other roles only with no administrative rights (WITH NO ADMIN OPTION clause). The WITH ADMIN OPTION and WITH ADMIN ONLY OPTION clauses are invalid for the SYS_SPATIAL_ADMIN_ROLE system role.

You can grant system privileges and roles to, and revoke them from, the SYS_SPATIAL_ADMIN_ROLE system role, including the default privilege.

Procedure

To grant the SYS_SPATIAL_ADMIN_ROLE system role, execute:

```
GRANT ROLE SYS_SPATIAL_ADMIN_ROLE TO <grantee [,...]>
```

Related Information

GRANT ROLE Statement [page 323]

1.2.2.9 Revoking a System Role

Revokes a system role from a user or role.

Prerequisites

Administrative privilege over the system role being revoked.

Procedure

To revoke a system role, execute:

```
REVOKE ROLE <role_name> FROM <grantee [,...]>
```

Example

This statement revokes the dbo system role entirely from Mary:

```
REVOKE ROLE dbo FROM Mary
```

Related Information

REVOKE ROLE Statement [page 340]

1.2.3 Compatibility Roles

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

You can grant, revoke, and under specific conditions, deleted 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.

See *Considerations When Upgrading from Pre-16.0 Releases > Understanding Role-Based Security After Upgrading from 15.x* in the SAP IQ Installation and Update Guide appropriate to your operating system.

1.2.4 Views, Procedures, and Tables That Are Owned by Roles

Views, procedures, and tables are more easily managed when they are owned by a user-extended role instead of a user.

To eliminate having to qualify the object name, make users who need access to a table, view, or stored procedure members of the role that owns the object.

For example, the table `Employees` is owned by the role `Personnel`, of which `Jeff` is a member. When `Jeff` wants to refer to the `Employees` table, he need only specify the name of the table in SQL statements, for example:

```
SELECT * FROM EMPLOYEES
```

However, when `John`, who is not a member of `Personnel`, wants to refer to the `Employees` table, he must use the qualified name of the table, for example:

```
SELECT * FROM PERSONNEL.EMPLOYEES
```

> **Note**
>
> Since ownership of database objects is associated with a single user ID, when the owner is a role, ownership of the table is not inherited by members of the role.

DO not grant system privileges to roles that own objects. Instead:

- Create distinct roles with specific system privileges granted
- Grant users who require the specific system privileges membership to the applicable role
- Grant each distinct role to the role that owns the object.

This allows for complete control of the tasks performed by each user. Maintain authorized tasks by granting and revoking membership in the applicable role associated with the object.

For example, the table `Sales` is owned by the `Sales1` role. Users `Mary`, `Bob`, `Joe`, `Laurel`, and `Sally` are granted membership to `Sales1`. Create `Task1_role` and granted it the system privileges necessary to complete a specific task. Grant `Task1_role` to `Mary` and `Bob`. Create `Task2_role`, grant it specific system privileges, and grant it to `Joe` and `Sally`. Finally, grant both `Task1_role` and `Task2_role` to `Sales1`. Though both roles are granted to `Sales1`, the underlying system privileges of `Task1_role` and `Task2_role` are not automatically inherited by the other members of `Sales1`. `Mary` and `Bob` can perform different tasks than `Joe` and `Sally`. Since `Laurel` has not been granted to either `Task1_role` or `Task2_role`, and no system privileges have been granted directly to `Sales1`, `Laurel` can perform no privileged tasks on the `Sales` table. This configuration allows you to maintain and control the tasks that can be performed by each user.

1.2.5 Display Roles Granted

The `sp_displayroles` stored procedure returns all roles that are granted to the specified system privilege, system role, user-defined role, or user name, or displays the entire hierarchy tree of roles.

The report includes role name, parent role name, type of grant (with or without administrative privilege), and the level of the role hierarchy.

No system privileges are required to execute `sp_displayroles` on your own user ID. To execute the procedure on other users requires the MANAGE ROLES system privilege. To execute the procedure for a role or system privilege requires administrative privilege over the role or system privilege specified.

Example

The example returns all roles granted to the user issuing the command.

```
CALL sp_displayroles();
```

This examples returns the list of system privileges granted to the SYS_SPATIAL_ADMIN_ROLE system role:

```
CALL sp_displayroles( 'SYS_SPATIAL_ADMIN_ROLE' );
```

role_name	parent_role_name	grant_type	role_level
MANAGE ANY SPATIAL OBJECT	(NULL)	NO ADMIN	1

This examples returns the list of system privileges granted to the SYS_SPATIAL_ADMIN_ROLE, including all roles above it in the hierarchy of roles:

```
CALL sp_displayroles( 'SYS_SPATIAL_ADMIN_ROLE', 'expand_up');
```

role_name	parent_role_name	grant_type	role_level
SYS_AUTH_DBA_ROLE	dbo	ADMIN	-3
SYS_AUTH_SSO_ROLE	SYS_AUTH_DBA_ROLE	ADMIN	-3
MANAGE ROLES	SYS_AUTH_REMOTE_DBA_ROLE	ADMIN	-2
MANAGE ROLES	SYS_AUTH_SSO_ROLE	ADMIN	-1
MANAGE ROLES	SYS_REPLICATION_ADMIN_ROLE	NO ADMIN	-1
SYS_SPATIAL_ADMIN_ROLE	MANAGE ROLES	ADMIN	0

Related Information

sp_displayroles System Procedure [page 378]

1.2.6 Determining the Roles and Privileges Granted to a User

The `sp_has_role` stored function returns an integer value that indicates whether the invoker of the procedure has been granted the specified system privilege or user-defined role.

No system privileges are required to execute the function. When used for permission checking within user-defined stored procedures, this function can display an error message when a user fails a permission check.

1 indicates the system privilege or user-defined role is granted to the invoking user.

0 or Permission denied: you do not have permission to execute this command/procedure indicates the system privilege or user-defined role is not granted to the invoking user. The error message replaces the value 0 when the throw_error argument is set to 1.

-1 indicates the system privilege or user-defined role specified does not exist. No error message appears, even if the throw_error argument is set to 1.

Related Information

SP_HAS_ROLE Function [System] [page 382]

1.3 Privileges

A privilege grants users the ability to perform an authorized operation on the system. For example, altering a table is a privileged operation, depending on the type of alteration you are making.

There are two types of privileges: system privileges and object-level privileges.

System privileges give you the general right to perform a privileged operation, while object-level privileges restrict you to performing the operation on a specific object. For example, if you have the ALTER ANY TABLE system privilege, you can alter any table in the system. If you have the ALTER TABLE system privilege, you can only alter tables you own, or tables on which you have been granted the ALTER object-level privilege. Object-level privileges can be granted or revoked, but not created or dropped.

System privileges are built in to the database and can be granted or revoked, but not created or dropped. With the exception of the MANAGE ROLES and UPGRADE ROLE privileges, system privileges cannot be modified. Each system privilege, with the exception of the SET USER system privilege, is granted by default to either the SYS_AUTH_SA_ROLE or SYS_AUTH_SSO_ROLE role, but not both. The SET USER system privilege is granted to both roles.

You grant and revoke system and object-level privileges using the `GRANT` and `REVOKE` statements.

In this section:

System Privileges [page 48]
System privileges let you control access to authorized system operations. Each privileged database task on the server requires specific system privileges. System privileges can be granted individually to users or roles.

Object-Level Privileges [page 105]
Database object-level privileges can be granted to and revoked from users.

System Procedure Privileges [page 122]
There are two security models under which privileged system procedures can run. Each model grants the ability to run the system procedure differently.

1.3.1 Privileges Versus Permissions

Permission and privilege do not mean the same thing in role-based security. A user may have been granted the privilege required to perform an authorized task, but not have the necessary permission to perform the authorized task on the required object.

A privilege grants a user or role the ability to perform a specific authorized task. Permission, however, refers to the context in which the task is being performed.

When performing an authorized task, if a failure occurs, the error message that appears often indicates that the user does not have permission to perform the task, not that the user does not have the privilege to perform the task. Before executing a privileged task or operation, the system verifies that the user has the correct privilege to perform the:

- Privileged operation
- Privileged operation on the acted-on object
- Privileged operation in the context in which he or she is attempting it

If the user does not have the correct privilege at any level, he or she is said to not have permission to perform the task. The operation fails and an error message appears.

Example

A user has been granted the ALTER privilege only on a text configuration object called `Myconfig.ini`.

Object privilege: The user attempts to alter a text configuration object other than `Myconfig.ini`. The task fails because the ALTER privilege granted to the user is specific to the`Myconfig.ini` `Myconfig.ini` text object, not any text object.

Context privilege: The user attempts to drop a prefilter on `Myconfig.ini`. Though the user has been granted the ALTER privilege on `Myconfig.ini`, to drop a prefilter on a text configuration object requires the ALTER ANY TEXT CONFIGURATION or ALTER ANY OBJECT system privilege, which has not been granted to the user.

1.3.2 System Privileges

System privileges let you control access to authorized system operations. Each privileged database task on the server requires specific system privileges. System privileges can be granted individually to users or roles.

When a system privilege is granted to a role, all members of the role inherit the system privilege. All new members of a role automatically inherit all of the underlying system privileges of a role.

Each system privilege, with the exception of the SET USER system privilege, by default, is granted to either the SYS_AUTH_SA_ROLE or the SYS_AUTH_SSO_ROLE role, but not both. The exception, SET USER system privilege, is granted to both roles.

Individually granting the underlying system privileges of a role is semantically equivalent to granting the role itself. You can grant system privileges to multiple user-defined system roles in any combination to meet the functional security requirements of your organization.

With the exception of MANAGE ROLES and UPGRADE ROLE, you cannot modify system privileges. System privileges can be granted to, and revoked from, roles and users, but they cannot be dropped. System privileges cannot own objects.

In this section:

1.3.2.1 System Privileges Listed by Functional Area

A list of system privileges organized by functional area.

In this section:

Data Type System Privileges [page 55]

System privileges pertaining to performing authorized tasks on data types.

Dbspaces System Privileges [page 56]

System privileges pertaining to performing authorized tasks on dbspaces.

Debugging System Privileges [page 57]

System privileges pertaining to performing authorized tasks related to debugging.

Events System Privileges [page 58]

System privileges pertaining to authorized tasks on events.

External Environment System Privileges [page 59]

System privileges pertaining to performing authorized tasks on external environments.

Files System Privileges [page 61]

System privileges pertaining to authorized tasks for files.

Indexes System Privileges [page 62]

System privileges pertaining to authorized tasks for indexes.

LDAP System Privileges [page 64]

System privileges pertaining to performing authorized tasks on an LDAP server configuration object.

Materialized Views System Privileges [page 65]

System privileges pertaining to performing authorized tasks on materialized views.

Messages System Privileges [page 67]

System privileges pertaining to performing authorized tasks for messages.

Miscellaneous System Privileges [page 68]

System privileges pertaining to performing miscellaneous authorized tasks.

Mirror Server System Privileges [page 73]

System privileges pertaining to authorized tasks for mirrored servers.

Multiplex System Privileges [page 74]

System privileges required to perform authorized tasks in a multiplex environment.

Procedures System Privileges [page 75]

System privileges pertaining to performing authorized tasks for procedures.

Replication System Privileges [page 78]

System privileges pertaining to performing authorized replication tasks.

Roles System Privileges [page 79]

System privileges pertaining to performing authorized tasks for roles.

Sequences System Privileges [page 81]

System privileges pertaining to performing authorized tasks for sequencing.

Server Operator System Privileges [page 83]

System privileges pertaining to performing authorized server operator tasks.

Spatial Objects System Privileges [page 84]

System privileges pertaining to performing authorized tasks on spatial objects.

Statistics System Privileges [page 85]

System privileges pertaining to performing authorized tasks on statistics.

Tables System Privileges [page 85]

System privileges pertaining to performing authorized tasks on tables.

Text Configurations System Privileges [page 90]
System privileges pertaining to performing authorized tasks on text configurations.

Triggers System Privileges [page 93]
System privileges pertaining to performing authorized task on triggers.

Users and Login Management System Privileges [page 94]
System privileges pertaining to performing authorized task on users and login policies.

Views System Privileges [page 96]
System privileges pertaining to performing authorized tasks on views.

Web Services System Privileges [page 98]
System privileges pertaining to performing authorized task on Web services.

1.3.2.1.1 Database System Privileges

System privileges pertaining to performing authorized tasks on databases.

In this section:

ALTER DATABASE System Privilege [page 51]
Required to alter a database.

BACKUP DATABASE System Privilege [page 51]
Allows a user to back up a database on one or more archive devices.

CHECKPOINT System Privilege [page 51]
Required to force the database server to execute a checkpoint.

DROP CONNECTION System Privilege [page 52]
Required to drop any user connections to the database.

MANAGE PROFILING System Privilege [page 52]
Required to enable or disable server tracing for application profiling. The DIAGNOSTICS system role is also required to fully utilize diagnostics functionality for user information.

MONITOR System Privilege [page 53]
Required to allow a user to perform monitoring related tasks such as access privileged statistics, run server monitor related procedures, and so on.

Related Information

List All System Privileges [page 99]

1.3.2.1.1.1 ALTER DATABASE System Privilege

Required to alter a database.

The ALTER DATABASE system privilege allows a user to:

- Perform a database upgrade
- Perform cost model calibration
- Load statistics
- Change transaction logs (also requires the SERVER OPERATOR system privilege)
- Change ownership of the database (also requires the MANAGE ANY MIRROR SERVER system privilege)

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.1.2 BACKUP DATABASE System Privilege

Allows a user to back up a database on one or more archive devices.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.1.3 CHECKPOINT System Privilege

Required to force the database server to execute a checkpoint.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.1.4 DROP CONNECTION System Privilege

Required to drop any user connections to the database.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.1.5 MANAGE PROFILING System Privilege

Required to enable or disable server tracing for application profiling. The DIAGNOSTICS system role is also required to fully utilize diagnostics functionality for user information.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.1.6 MONITOR System Privilege

Required to allow a user to perform monitoring related tasks such as access privileged statistics, run server monitor related procedures, and so on.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.2 Database Options System Privileges

System privileges pertaining to performing authorized tasks to set database options.

In this section:

- SET ANY PUBLIC OPTION System Privilege [page 54]
  Required to set any PUBLIC system database option that does not require the SET ANY SECURITY OPTION or SET ANY SYSTEM OPTION system privileges.
- SET ANY SECURITY OPTION System Privilege [page 54]
  Required to set any PUBLIC security database options.
- SET ANY SYSTEM OPTION System Privilege [page 54]
  Required to set any PUBLIC system database options.
- SET ANY USER DEFINED OPTION System Privilege [page 55]
  Required to set any user-defined options.

Related Information

List All System Privileges [page 99]

1.3.2.1.2.1 SET ANY PUBLIC OPTION System Privilege

Required to set any PUBLIC system database option that does not require the SET ANY SECURITY OPTION or SET ANY SYSTEM OPTION system privileges.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.2.2 SET ANY SECURITY OPTION System Privilege

Required to set any PUBLIC security database options.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.2.3 SET ANY SYSTEM OPTION System Privilege

Required to set any PUBLIC system database options.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]

List All System Privileges [page 99]

1.3.2.1.2.4 SET ANY USER DEFINED OPTION System Privilege

Required to set any user-defined options.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.3 Data Type System Privileges

System privileges pertaining to performing authorized tasks on data types.

In this section:

- ALTER DATATYPE System Privilege [page 55]
  Required to alter data types.
- CREATE DATATYPE System Privilege [page 56]
  Required to create data types.
- DROP DATATYPE System Privilege [page 56]
  Required to drop data types.

Related Information

List All System Privileges [page 99]

1.3.2.1.3.1 ALTER DATATYPE System Privilege

Required to alter data types.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.3.2 CREATE DATATYPE System Privilege

Required to create data types.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.3.3 DROP DATATYPE System Privilege

Required to drop data types.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.4 Dbspaces System Privileges

System privileges pertaining to performing authorized tasks on dbspaces.

In this section:

MANAGE ANY DBSPACE System Privilege [page 57]
Required to perform management-related tasks on dbspaces.

Related Information

List All System Privileges [page 99]

1.3.2.1.4.1 MANAGE ANY DBSPACE System Privilege

Required to perform management-related tasks on dbspaces.

The MANAGE ANY DBSPACE system privilege allows a user to:

- Issue CREATE, ALTER, DROP, or COMMENT statements on any dbspace
- GRANT or REVOKE the CREATE object-level privilege on any dbspace
- Move data to any dbspace
- Issue a read-only selective restore statement on any dbspace
- Run the database delete file function

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.5 Debugging System Privileges

System privileges pertaining to performing authorized tasks related to debugging.

In this section:

DEBUG ANY PROCEDURE System Privilege [page 58]
Required to debug all code in any database object.

Related Information

List All System Privileges [page 99]

1.3.2.1.5.1 DEBUG ANY PROCEDURE System Privilege

Required to debug all code in any database object.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.6 Events System Privileges

System privileges pertaining to authorized tasks on events.

In this section:

MANAGE ANY EVENT System Privilege [page 58]
Required to create, alter, drop, or trigger events.

Related Information

List All System Privileges [page 99]

1.3.2.1.6.1 MANAGE ANY EVENT System Privilege

Required to create, alter, drop, or trigger events.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.7 External Environment System Privileges

System privileges pertaining to performing authorized tasks on external environments.

In this section:

Related Information

List All System Privileges [page 99]

1.3.2.1.7.1 CREATE EXTERNAL REFERENCE System Privilege

Required to create external references in the database.

This system privilege is required in addition to any other system privileges required for creating a database object that references an external object.

For example, to create an external:

- Term breaker or a self-owned text configuration that uses an external term breaker requires the system privilege CREATE TEXT CONFIGURATION in addition to the CREATE EXTERNAL REFERENCE system privilege.
- Procedure or function requires the CREATE PROCEDURE system privilege in addition to the CREATE EXTERNAL REFERENCE system privilege.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.7.2 MANAGE ANY EXTERNAL ENVIRONMENT System Privilege

Required to manage external environments.

The MANAGE ANY EXTERNAL ENVIRONMENT system privilege allows a user to issue ALTER, COMMENT, START or STOP statements on an external environment.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.7.3 MANAGE ANY EXTERNAL OBJECT System Privilege

Required to issue INSTALL, COMMENT ON, or REMOVE EXTERNAL OBJECT statements.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.8 Files System Privileges

System privileges pertaining to authorized tasks for files.

In this section:

Related Information

List All System Privileges [page 99]

1.3.2.1.8.1 READ CLIENT FILE System Privilege

Required to read a file resident on the client machine.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.8.2 READ FILE System Privilege

Required to read a file resident on the server machine.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.8.3 WRITE CLIENT FILE System Privilege

Required to write a file resident on the client machine.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.8.4 WRITE FILE System Privilege

Required to write a file resident on the server machine.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.9 Indexes System Privileges

System privileges pertaining to authorized tasks for indexes.

In this section:

ALTER ANY INDEX System Privilege [page 63]
Required to alter an existing index.

CREATE ANY INDEX System Privilege [page 63]
Required to create a new index.

DROP ANY INDEX System Privilege [page 64]
Required to drop indexes on any table owned by any user.

Related Information

List All System Privileges [page 99]

1.3.2.1.9.1 ALTER ANY INDEX System Privilege

Required to alter an existing index.

The ALTER ANY INDEX system privilege allows a user to:

- Alter indexes on any table owned by any user
- Issue the COMMENT statement on any index owned by any user

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.9.2 CREATE ANY INDEX System Privilege

Required to create a new index.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

The CREATE ANY INDEX system privilege allows a user to:

- Create indexes on any table owned by any user
- Issue COMMENT statement on any index owned by any user

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.9.3 DROP ANY INDEX System Privilege

Required to drop indexes on any table owned by any user.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.10 LDAP System Privileges

System privileges pertaining to performing authorized tasks on an LDAP server configuration object.

In this section:

MANAGE ANY LDAP SERVER System Privilege [page 64]
Required to issue CREATE, ALTER, or DROP statements on an LDAP server configuration object.

Related Information

List All System Privileges [page 99]

1.3.2.1.10.1 MANAGE ANY LDAP SERVER System Privilege

Required to issue CREATE, ALTER, or DROP statements on an LDAP server configuration object.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.11 Materialized Views System Privileges

System privileges pertaining to performing authorized tasks on materialized views.

In this section:

- CREATE ANY MATERIALIZED VIEW System Privilege [page 65]
  Required to create materialized views that are owned by any user. It also allows users to issue the COMMENT statement on materialized views owned by any user.
- CREATE MATERIALIZED VIEW System Privilege [page 66]
  Required to create self-owned materialized views. It also allows users to issue the COMMENT statement on self-owned materialized views.
- ALTER ANY MATERIALIZED VIEW System Privilege [page 66]
  Required to alter materialized views owned by any user. It also allows users to issue the COMMENT statement on materialized views owned by any user.
- DROP ANY MATERIALIZED VIEW System Privilege [page 67]
  Required to drop materialized views owned by any user.

Related Information

List All System Privileges [page 99]

1.3.2.1.11.1 CREATE ANY MATERIALIZED VIEW System Privilege

Required to create materialized views that are owned by any user. It also allows users to issue the COMMENT statement on materialized views owned by any user.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.11.2 CREATE MATERIALIZED VIEW System Privilege

Required to create self-owned materialized views. It also allows users to issue the COMMENT statement on self-owned materialized views.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.11.3 ALTER ANY MATERIALIZED VIEW System Privilege

Required to alter materialized views owned by any user. It also allows users to issue the COMMENT statement on materialized views owned by any user.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.11.4 DROP ANY MATERIALIZED VIEW System Privilege

Required to drop materialized views owned by any user.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.12 Messages System Privileges

System privileges pertaining to performing authorized tasks for messages.

In this section:

CREATE MESSAGE System Privilege [page 67]
Required to create messages.

DROP MESSAGE System Privilege [page 68]
Required to drop messages.

Related Information

List All System Privileges [page 99]

1.3.2.1.12.1 CREATE MESSAGE System Privilege

Required to create messages.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]

REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.12.2 DROP MESSAGE System Privilege

Required to drop messages.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.13 Miscellaneous System Privileges

System privileges pertaining to performing miscellaneous authorized tasks.

In this section:

Related Information

List All System Privileges [page 99]

1.3.2.1.13.1 ALTER ANY OBJECT System Privilege

Required to alter an object owned by anyone.

The ALTER ANY OBJECT system privilege allows a user to issue these statements:

- ALTER TABLE
- ALTER INDEX
- ALTER JOIN INDEX
- ALTER VIEW
- ALTER MATERIALIZED VIEW
- ALTER PROCEDURE
- ALTER EVENT
- ALTER SEQUENCE
- ALTER FUNCTION
- ALTER DATATYPE
- ALTER MESSAGE
- ALTER TEXT CONFIGURATION
- ALTER TRIGGER
- ALTER STATISTICS
- COMMENT on different objects
- ALTER SPATIAL REFERENCE SYSTEM
- ALTER SPATIAL UNIT OF MEASURE

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.13.2 ALTER ANY OBJECT OWNER System Privilege

Required to change the owner of a user table owned by anyone.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

> i Note
>
> This system privilege applies to table objects only. Owners of other objects, such as procedures, materialized views, and so on, cannot be changed.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.13.3 COMMENT ANY OBJECT System Privilege

Required to comment on any object owned by any user.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.13.4 CREATE ANY OBJECT System Privilege

Required to create an object owned by anyone.

The CREATE ANY OBJECT system privilege allows a user to issue these statements:

- COMMENT on different objects
- CREATE DATATYPE
- CREATE EVENT

- CREATE FUNCTION
- CREATE INDEX
- CREATE JOIN INDEX
- CREATE MATERIALIZED VIEW
- CREATE MESSAGE
- CREATE PROCEDURE
- CREATE SCHEMA
- CREATE SEQUENCE
- CREATE SPATIAL REFERENCE SYSTEM
- CREATE SPATIAL UNIT OF MEASURE
- CREATE STATISTICS
- CREATE TABLE
- CREATE TEXT CONFIGURATION
- CREATE VIEW

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.13.5 DROP ANY OBJECT System Privilege

Required to drop an object owned by anyone.

The DROP ANY OBJECT system privilege allows a user to issue these statements:

- DROP DATATYPE
- DROP EVENT
- DROP FUNCTION
- DROP INDEX
- DROP JOIN INDEX
- DROP MATERIALIZED VIEW
- DROP MESSAGE
- DROP PROCEDURE
- DROP SEQUENCE
- DROP SPATIAL REFERENCE SYSTEM
- DROP SPATIAL UNIT OF MEASURE
- DROP STATISTICS
- DROP TABLE

- DROP TEXT CONFIGURATION
- DROP TRIGGER
- DROP VIEW

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.13.6 MANAGE ANY OBJECT PRIVILEGES System Privilege

Required to manage objects.

The MANAGE ANY OBJECT PRIVILEGES system privilege allows a user to perform management-related tasks, such as:

- Grant any object-level privilege (INSERT, UPDATE, DELETE, SELECT, ALTER, REFERENCES or EXECUTE) on objects owned by any user
- Revoke any object-level privilege granted by the object owner or another user with MANAGE ANY OBJECT PRIVILEGES system privilege

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.13.7 REORGANIZE ANY OBJECT System Privilege

Required to issue the REORGANIZE statement on applicable objects owned by any user.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.13.8 VALIDATE ANY OBJECT System Privilege

Required to validate or check tables, materialized views, indexes, or databases in the system store owned by any user.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.14 Mirror Server System Privileges

System privileges pertaining to authorized tasks for mirrored servers.

In this section:

MANAGE ANY MIRROR SERVER System Privilege [page 73]
Required to perform high availability server administrative tasks.

Related Information

List All System Privileges [page 99]

1.3.2.1.14.1 MANAGE ANY MIRROR SERVER System Privilege

Required to perform high availability server administrative tasks.

The MANAGE ANY MIRROR SERVER system privilege allows a user to:

- Issue the CREATE, ALTER or DROP statement on mirrored servers
- Change mirror server parameters
- Set options on mirror servers
- Execute the ALTER statement to change ownership of a database

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.15 Multiplex System Privileges

System privileges required to perform authorized tasks in a multiplex environment.

In this section:

ACCESS SERVER LS System Privilege [page 74]
Allows logical server connection using the SERVER logical server context.

MANAGE MULTIPLEX System Privilege [page 75]
Allows administrative tasks related to multiplex server management.

Related Information

List All System Privileges [page 99]

1.3.2.1.15.1 ACCESS SERVER LS System Privilege

Allows logical server connection using the SERVER logical server context.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]

REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.15.2 MANAGE MULTIPLEX System Privilege

Allows administrative tasks related to multiplex server management.

The MANAGE MULTIPLEX system privilege allows a user to:

- Issue multiplex-related CREATE, ALTER, DROP, or COMMENT statements on logical server policies
- Issue multiplex-related CREATE, ALTER, DROP, or COMMENT statements on logical servers
- Perform exclusive assignment of a dbspace to logical servers
- Release a populated dbspace from the exclusive use of a logical server

> i Note
>
> The MANAGE MULTIPLEX system privilege also manages failover configurations, and is required for a manual failover.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.16 Procedures System Privileges

System privileges pertaining to performing authorized tasks for procedures.

In this section:

EXECUTE ANY PROCEDURE System Privilege [page 78]
Required to execute any stored procedure or function owned by any user.

MANAGE AUDITING System Privilege [page 78]
Required to run the `sa_audit_string` stored procedure.

Related Information

List All System Privileges [page 99]

1.3.2.1.16.1 ALTER ANY PROCEDURE System Privilege

Required to alter any stored procedure or function owned by any user.

The ALTER ANY PROCEDURE system privilege allows a user to:

- Alter stored procedures and functions owned by any user
- Issue the COMMENT statement on procedures owned by any user

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.16.2 CREATE ANY PROCEDURE System Privilege

Required to create any stored procedure or function owned by any user.

The CREATE ANY PROCEDURE system privilege allows a user to:

- Create stored procedures and functions owned by any user
- Issue the COMMENT statement on procedures owned by any user

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.16.3 CREATE PROCEDURE System Privilege

Required to create a self-owned stored procedure or function.

The CREATE PROCEDURE system privilege allows a user to:

- Create self-owned stored procedures and functions
- Issue the COMMENT statement on self-owned procedures

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.16.4 DROP ANY PROCEDURE System Privilege

Required to drop any stored procedure or function owned by any user.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.16.5 EXECUTE ANY PROCEDURE System Privilege

Required to execute any stored procedure or function owned by any user.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.16.6 MANAGE AUDITING System Privilege

Required to run the `sa_audit_string` stored procedure.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.17 Replication System Privileges

System privileges pertaining to performing authorized replication tasks.

In this section:

Related Information

List All System Privileges [page 99]

1.3.2.1.17.1 MANAGE REPLICATION System Privilege

Required to perform replication-related tasks.

The MANAGE REPLICATION system privilege allows a user to:

- Issue the CREATE, ALTER, DROP, or COMMENT PUBLICATION statement
- Issue the CREATE, ALTER, DROP, or SYNCHRONIZATION SUBSCRIPTION statement
- Issue the CREATE, ALTER, DROP, or SYNCHRONIZATION USER statement
- Issue the CREATE, ALTER, DROP, or COMMENT SYNCHRONIZATION PROFILE statement
- Issue the CREATE or DROP SUBSCRIPTION statement
- Issue the CREATE REMOTE MESSAGE TYPE statement
- Issue the DROP REMOTE MESSAGE TYPE statement
- Issue the GRANT or REVOKE CONSOLIDATE statement
- Issue the GRANT or REVOKE REMOTE statement
- Issue the GRANT or REVOKE PUBLISH statement
- Issue the LOCK FEATURE statement
- Issue the START, STOP, or SYNCHRONIZE SUBSCRIPTION statement
- Issue the PASSSTHROUGH statement
- Issue the REMOTE RESET statement
- Issue the SET REMOTE OPTION statement
- Issue the START or STOP SYNCHRONIZATION SCHEMA CHANGE statement
- Issue the SYNCHRONIZE PROFILE statement
- Execute the SA_SETREMOTEUSER procedure
- Execute the SA_SETSUBSCRIPTION procedure

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.18 Roles System Privileges

System privileges pertaining to performing authorized tasks for roles.

In this section:

Required to administrate new system privileges introduced when upgrading an IQ database earlier than 16.0.

Related Information

List All System Privileges [page 99]

1.3.2.1.18.1 MANAGE ROLES System Privilege

Required to create new roles and act as the default administrator of roles.

While the MANAGE ROLES system privilege allows a user to create a new user-defined role, it does not allow him or her to delete the role. For this, a user requires administrative rights on the role.

Users who are granted the MANAGE ROLES system privilege serve as default global role administrators on a user-defined role.

If no role administrator is specified during the role creation process, the MANAGE ROLES system privilege (SYS_MANAGE_ROLES_ROLE) is automatically granted to the role with the ADMIN ONLY OPTION clause, which allows the global role administrator to administer the role. If at least one role administrator is specified during the creation process, the MANAGE ROLES system privilege is not granted to the role, and global role administrators cannot manage the role.

MANAGE ROLES is the only system privilege that can be granted the ability to administer user-defined roles.

> i Note
>
> You can also grant role administration directly to users either while creating a role or, or after. When role administration is granted directly to a user, the user does not require the MANAGE ROLES system privilege to administer the role.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.18.2 UPGRADE ROLE System Privilege

Required to administrate new system privileges introduced when upgrading an IQ database earlier than 16.0.

By default, the UPGRADE ROLE system privilege is granted to the SYS_AUTH_SA_ROLE role, if it exists.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.19 Sequences System Privileges

System privileges pertaining to performing authorized tasks for sequencing.

In this section:

ALTER ANY SEQUENCE System Privilege [page 81]
    Required to alter any sequence.

CREATE ANY SEQUENCE System Privilege [page 82]
    Required to create any sequence.

DROP ANY SEQUENCE System Privilege [page 82]
    Required to drop any sequence.

USE ANY SEQUENCE System Privilege [page 82]
    Required to use any sequence.

Related Information

List All System Privileges [page 99]

1.3.2.1.19.1 ALTER ANY SEQUENCE System Privilege

Required to alter any sequence.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]

REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.19.2 CREATE ANY SEQUENCE System Privilege

Required to create any sequence.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.19.3 DROP ANY SEQUENCE System Privilege

Required to drop any sequence.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.19.4 USE ANY SEQUENCE System Privilege

Required to use any sequence.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.20 Server Operator System Privileges

System privileges pertaining to performing authorized server operator tasks.

In this section:

SERVER OPERATOR System Privilege [page 83]
Required to perform server-operator tasks.

Related Information

List All System Privileges [page 99]

1.3.2.1.20.1 SERVER OPERATOR System Privilege

Required to perform server-operator tasks.

The SERVER OPERATOR system privilege allows a user to:

- Create databases
- Cache management
- Drop databases
- Start or stop a database
- Start or stop a database engine
- Create, alter, or drop a server
- Create encrypted or decrypted databases
- Create encrypted or decrypted files
- Issue `ALTER` statement to change transaction logs on a database
- Issue `RESTORE DATABASE` statement for a full database restore or to restore the catalog only

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.21 Spatial Objects System Privileges

System privileges pertaining to performing authorized tasks on spatial objects.

In this section:

MANAGE ANY SPATIAL OBJECT System Privilege [page 84]
Required to manage any spatial objects.

Related Information

List All System Privileges [page 99]

1.3.2.1.21.1 MANAGE ANY SPATIAL OBJECT System Privilege

Required to manage any spatial objects.

The MANAGE ANY SPATIAL OBJECT system privilege allows a user to issue:

- CREATE, ALTER, or DROP statements on spatial objects
- CREATE, ALTER, or DROP statements on spatial units of measure
- COMMENT statements on spatial units of measure.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.22 Statistics System Privileges

System privileges pertaining to performing authorized tasks on statistics.

In this section:

MANAGE ANY STATISTICS System Privilege [page 85]
Required to issue CREATE, ALTER, DROP, or UPDATE statements on statistics for any table.

Related Information

List All System Privileges [page 99]

1.3.2.1.22.1 MANAGE ANY STATISTICS System Privilege

Required to issue CREATE, ALTER, DROP, or UPDATE statements on statistics for any table.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.23 Tables System Privileges

System privileges pertaining to performing authorized tasks on tables.

In this section:

ALTER ANY TABLE System Privilege [page 86]
Required to alter any table owned by anyone.

CREATE ANY TABLE System Privilege [page 87]
Required to create tables owned by any user.

CREATE PROXY TABLE System Privilege [page 87]
Required to create self-owned proxy tables.

CREATE TABLE System Privilege [page 87]
Required to create self-owned tables.

DELETE ANY TABLE System Privilege [page 88]
Required to delete rows from tables, table partitions, or views owned by any user.

DROP ANY TABLE System Privilege [page 88]
Required to drop tables owned by any user.

INSERT ANY TABLE System Privilege [page 89]
Required to insert rows into tables and views owned by anyone.

LOAD ANY TABLE System Privilege [page 89]
Required to execute LOAD command for any table where the `-gl` server switch is set to DBA.

SELECT ANY TABLE System Privilege [page 89]
Required to query tables, views, or materialized views owned by any user.

TRUNCATE ANY TABLE System Privilege [page 90]
Required to execute TRUNCATE command for any table.

UPDATE ANY TABLE System Privilege [page 90]
Required to update rows in tables and views owned by any user.

Related Information

List All System Privileges [page 99]

1.3.2.1.23.1 ALTER ANY TABLE System Privilege

Required to alter any table owned by anyone.

The ALTER DATABASE system privilege allows a user to:

- Issue ALTER or TRUNCATE statements on tables, table partitions, or views owned by any user
- Issue the COMMENT statement on tables owned by any user
- Issue the COMMENT statement on columns on tables owned by any user

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.23.2 CREATE ANY TABLE System Privilege

Required to create tables owned by any user.

The CREATE ANY TABLE system privilege allows a user to:

- Create tables, including proxy tables, owned by any user
- Issue the COMMENT statement on tables owned by any user
- Issue the COMMENT statement on columns on tables owned by any user

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.23.3 CREATE PROXY TABLE System Privilege

Required to create self-owned proxy tables.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.23.4 CREATE TABLE System Privilege

Required to create self-owned tables.

The CREATE TABLE system privilege allows a user to:

- Create self-owned tables except proxy tables
- Issue the COMMENT statement on self-owned tables
- Issue the COMMENT statement on columns on self-owned tables

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.23.5 DELETE ANY TABLE System Privilege

Required to delete rows from tables, table partitions, or views owned by any user.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.23.6 DROP ANY TABLE System Privilege

Required to drop tables owned by any user.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.23.7 INSERT ANY TABLE System Privilege

Required to insert rows into tables and views owned by anyone.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.23.8 LOAD ANY TABLE System Privilege

Required to execute LOAD command for any table where the `-gl` server switch is set to DBA.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.23.9 SELECT ANY TABLE System Privilege

Required to query tables, views, or materialized views owned by any user.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.23.10 TRUNCATE ANY TABLE System Privilege

Required to execute TRUNCATE command for any table.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.23.11 UPDATE ANY TABLE System Privilege

Required to update rows in tables and views owned by any user.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.24 Text Configurations System Privileges

System privileges pertaining to performing authorized tasks on text configurations.

In this section:

Required to drop text configurations owned by any user.

Related Information

List All System Privileges [page 99]

1.3.2.1.24.1 ALTER ANY TEXT CONFIGURATION System Privilege

Required to alter text configurations owned by any user.

The ALTER ANY TEXT CONFIGURATION system privilege allows a user to issue:

- ALTER statements on text configurations owned by any user
- COMMENT statements on text configurations owned by any user

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.24.2 CREATE ANY TEXT CONFIGURATION System Privilege

Required to create text configurations owned by other users.

The CREATE ANY TEXT CONFIGURATION system privilege allows a user to:

- Create configurations owned by any user
- Issue COMMENT statement on text configuration owned by any user

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.24.3 CREATE TEXT CONFIGURATION System Privilege

Required to create self owned text configurations.

The CREATE TEXT CONFIGURATION system privilege allows a user to:

- Create self owned text configurations
- Issue the COMMENT statement on self owned text configurations

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.24.4 DROP ANY TEXT CONFIGURATION System Privilege

Required to drop text configurations owned by any user.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.25 Triggers System Privileges

System privileges pertaining to performing authorized task on triggers.

In this section:

Related Information

List All System Privileges [page 99]

1.3.2.1.25.1 ALTER ANY TRIGGER System Privilege

Required to alter triggers. Users can also issue a COMMENT statement on tables if he or she has the ALTER privilege on the table.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.25.2 CREATE ANY TRIGGER System Privilege

Required to create triggers. Users can also issue a COMMENT statement on tables if they have the ALTER privilege on the table.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.26 Users and Login Management System Privileges

System privileges pertaining to performing authorized task on users and login policies.

In this section:

CHANGE PASSWORD System Privilege [page 94]
Allows users to manage passwords other than their own.

MANAGE ANY LOGIN POLICY System Privilege [page 95]
Required to manage login policies.

MANAGE ANY USER System Privilege [page 95]
Required to manage users.

SET USER System Privilege [page 96]
Allows a user to temporarily assume (impersonate) the specific roles and system privileges of another user.

Related Information

List All System Privileges [page 99]

1.3.2.1.26.1 CHANGE PASSWORD System Privilege

Allows users to manage passwords other than their own.

You can limit this system privilege to allow a user to manage passwords for a specific list of users, to manage passwords for any user granted a specific list of roles, or to manage passwords for any existing database user. Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

Passwords [page 126]
GRANT CHANGE PASSWORD Statement [page 316]

REVOKE CHANGE PASSWORD Statement [page 333]
List All System Privileges [page 99]

1.3.2.1.26.2 MANAGE ANY LOGIN POLICY System Privilege

Required to manage login policies.

The MANAGE ANY LOGIN POLICY system privilege allows a user to issue:

- CREATE, ALTER, or DROP statements on login policies
- COMMENT statements on login policies

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.26.3 MANAGE ANY USER System Privilege

Required to manage users.

The MANAGE ANY USER system privilege allows a user to:

- Issue CREATE, ALTER, or DROP statements on database users (including assigning initial password)
- Define authentication mechanisms for users (Kerberos, Integrated login)
- Issue CREATE or DROP statements on external logins
- Force password change on next login for any user
- Assign a login policy to any user
- Reset the login policy of any user
- Issue the COMMENT statement on users, integrated logins, or Kerberos logins

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]

List All System Privileges [page 99]

1.3.2.1.26.4 SET USER System Privilege

Allows a user to temporarily assume (impersonate) the specific roles and system privileges of another user.

> i Note
>
> The SET USER system privilege is two words; the SETUSER statement is one word.

When you grant the SET USER system privilege, you can define the scope of impersonation as:

- Any user in the database.
- Any user within a specified list of users (`<target_users_list>`).
- Any user who is a member of one or more of the specified roles (`<target_roles_list>`).

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

Impersonation [page 134]
GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.27 Views System Privileges

System privileges pertaining to performing authorized tasks on views.

In this section:

ALTER ANY VIEW System Privilege [page 97]
Required to alter views owned by any user.

CREATE ANY VIEW System Privilege [page 97]
Required to create views owned by any user.

CREATE VIEW System Privilege [page 98]
Required to create self-owned views.

DROP ANY VIEW System Privilege [page 98]
Required to drop a view owned by any user.

Related Information

List All System Privileges [page 99]

1.3.2.1.27.1 ALTER ANY VIEW System Privilege

Required to alter views owned by any user.

The ALTER ANY VIEW system privilege allows a user to:

- Alter views owned by any user
- Issue the COMMENT statement on views owned by any user

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.27.2 CREATE ANY VIEW System Privilege

Required to create views owned by any user.

The CREATE ANY VIEW system privilege allows a user to:

- Create views owned by any user
- Issue the COMMENT statement on views owned by any user

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.27.3 CREATE VIEW System Privilege

Required to create self-owned views.

The CREATE VIEW system privilege allows a user to:

- Create self-owned views
- Issue the COMMENT statement on self-owned views

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.27.4 DROP ANY VIEW System Privilege

Required to drop a view owned by any user.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.1.28 Web Services System Privileges

System privileges pertaining to performing authorized task on Web services.

In this section:

Related Information

List All System Privileges [page 99]

1.3.2.1.28.1 MANAGE ANY WEB SERVICE System Privilege

Required to manage tasks related to Web services.

The MANAGE ANY WEB SERVICE system privilege allows a user to issue:

- CREATE, ALTER, or DROP statements on Web services
- COMMENT statements on Web services

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Related Information

GRANT System Privilege Statement [page 329]
REVOKE System Privilege Statement [page 344]
List All System Privileges [page 99]

1.3.2.2 List All System Privileges

List of all system privileges.

System privileges control the rights of users to perform authorized database tasks.

Related Information

ACCESS SERVER LS System Privilege [page 74]
ALTER ANY INDEX System Privilege [page 63]
ALTER ANY MATERIALIZED VIEW System Privilege [page 66]
ALTER ANY OBJECT System Privilege [page 69]
ALTER ANY OBJECT OWNER System Privilege [page 70]
ALTER ANY PROCEDURE System Privilege [page 76]
ALTER ANY SEQUENCE System Privilege [page 81]
ALTER ANY TABLE System Privilege [page 86]
ALTER ANY TEXT CONFIGURATION System Privilege [page 91]
ALTER ANY TRIGGER System Privilege [page 93]

ALTER ANY VIEW System Privilege [page 97]

ALTER DATABASE System Privilege [page 51]

ALTER DATATYPE System Privilege [page 55]

BACKUP DATABASE System Privilege [page 51]

CHANGE PASSWORD System Privilege [page 94]

CHECKPOINT System Privilege [page 51]

COMMENT ANY OBJECT System Privilege [page 70]

CREATE ANY INDEX System Privilege [page 63]

CREATE ANY MATERIALIZED VIEW System Privilege [page 65]

CREATE ANY OBJECT System Privilege [page 70]

CREATE ANY PROCEDURE System Privilege [page 76]

CREATE ANY SEQUENCE System Privilege [page 82]

CREATE ANY TABLE System Privilege [page 87]

CREATE ANY TEXT CONFIGURATION System Privilege [page 91]

CREATE ANY TRIGGER System Privilege [page 93]

CREATE ANY VIEW System Privilege [page 97]

CREATE DATATYPE System Privilege [page 56]

CREATE EXTERNAL REFERENCE System Privilege [page 59]

CREATE MATERIALIZED VIEW System Privilege [page 66]

CREATE MESSAGE System Privilege [page 67]

CREATE PROCEDURE System Privilege [page 77]

CREATE PROXY TABLE System Privilege [page 87]

CREATE TABLE System Privilege [page 87]

CREATE TEXT CONFIGURATION System Privilege [page 92]

CREATE VIEW System Privilege [page 98]

DEBUG ANY PROCEDURE System Privilege [page 58]

DELETE ANY TABLE System Privilege [page 88]

DROP ANY INDEX System Privilege [page 64]

DROP ANY MATERIALIZED VIEW System Privilege [page 67]

DROP ANY OBJECT System Privilege [page 71]

DROP ANY PROCEDURE System Privilege [page 77]

DROP ANY SEQUENCE System Privilege [page 82]

DROP ANY TABLE System Privilege [page 88]

DROP ANY TEXT CONFIGURATION System Privilege [page 92]

DROP ANY VIEW System Privilege [page 98]

DROP CONNECTION System Privilege [page 52]

DROP DATATYPE System Privilege [page 56]

DROP MESSAGE System Privilege [page 68]

EXECUTE ANY PROCEDURE System Privilege [page 78]

LOAD ANY TABLE System Privilege [page 89]

INSERT ANY TABLE System Privilege [page 89]

MANAGE ANY DBSPACE System Privilege [page 57]

MANAGE ANY EVENT System Privilege [page 58]

MANAGE ANY EXTERNAL ENVIRONMENT System Privilege [page 60]

MANAGE ANY EXTERNAL OBJECT System Privilege [page 60]

MANAGE ANY LDAP SERVER System Privilege [page 64]

MANAGE ANY LOGIN POLICY System Privilege [page 95]

MANAGE ANY MIRROR SERVER System Privilege [page 73]

MANAGE ANY OBJECT PRIVILEGES System Privilege [page 72]

MANAGE ANY SPATIAL OBJECT System Privilege [page 84]

MANAGE ANY STATISTICS System Privilege [page 85]

MANAGE ANY USER System Privilege [page 95]

MANAGE ANY WEB SERVICE System Privilege [page 99]

MANAGE AUDITING System Privilege [page 78]

MANAGE MULTIPLEX System Privilege [page 75]

MANAGE PROFILING System Privilege [page 52]

MANAGE REPLICATION System Privilege [page 79]

MANAGE ROLES System Privilege [page 80]

MONITOR System Privilege [page 53]

READ CLIENT FILE System Privilege [page 61]

READ FILE System Privilege [page 61]

REORGANIZE ANY OBJECT System Privilege [page 72]

SELECT ANY TABLE System Privilege [page 89]

SERVER OPERATOR System Privilege [page 83]

SET ANY PUBLIC OPTION System Privilege [page 54]

SET ANY SECURITY OPTION System Privilege [page 54]

SET ANY SYSTEM OPTION System Privilege [page 54]

SET ANY USER DEFINED OPTION System Privilege [page 55]

SET USER System Privilege [page 96]

TRUNCATE ANY TABLE System Privilege [page 90]

UPDATE ANY TABLE System Privilege [page 90]

UPGRADE ROLE System Privilege [page 80]

USE ANY SEQUENCE System Privilege [page 82]

VALIDATE ANY OBJECT System Privilege [page 73]

WRITE CLIENT FILE System Privilege [page 62]

WRITE FILE System Privilege [page 62]

1.3.2.3 Granting a System Privilege to a User

Allow the granting of specific system privileges to specific users, with or without administrative rights.

Prerequisites

Administrative privilege over the system privilege being granted.

Context

> ⚠ Caution
>
> The syntax to grant a system privilege is the same for all system privileges except the CHANGE PASSWORD and SET USER system privileges.

Grant this system privilege using the WITH ADMIN OPTION, WITH NO ADMIN OPTION, or WITH ADMIN ONLY OPTION clause. If you do not specify a clause, the default is WITH NO ADMIN OPTION.

Procedure

To grant a system privilege to a user, execute one of these statements:

Administrative Option	Statement

With full administrative rights	`GRANT <system_privilege> TO <grantee [,...]>`
	`WITH ADMIN OPTION`

With administrative rights only	`GRANT <system_privilege> TO <grantee [,...]>`
	`WITH ADMIN ONLY OPTION`

With no administrative rights	`GRANT <system_privilege> TO <grantee [,...]>`
	`WITH NO ADMIN OPTION`

Related Information

GRANT System Privilege Statement [page 329]
GRANT CHANGE PASSWORD Statement [page 316]
GRANT SET USER Statement [page 327]

1.3.2.4 Revoking a System Privilege from a User

Revoke a specific system privilege and the right to administer the system privilege from specific users.

Prerequisites

Administrative privilege over the system privilege being revoked.

Context

> ⚠ Caution
>
> The syntax to revoke a system privilege applies to all system privileges except the CHANGE PASSWORD and SET USER system privileges.

Procedure

To revoke a system privilege from a user, execute one of these statements:

Administrative Option	Statement

Administrative rights only	`REVOKE ADMIN OPTION FOR <system_privilege>`
	`FROM <grantee [,...]>`

System privilege and any administrative rights	`REVOKE <system_privilege>`
	`FROM <grantee [,...]>`

Example

Assuming `Mary` and `Joe` were originally granted the BACKUP DATABASE system privilege with administrative rights, execute this statement to remove `Mary's` administrative rights to the system privilege only, leaving her ability to use the system privilege:

```
REVOKE ADMIN OPTION FOR BACKUP DATABASE FROM Mary
```

Execute this statement to remove the system privilege itself and all administrative rights from `Joe`:

```
REVOKE BACKUP DATABASE FROM Joe
```

Related Information

REVOKE System Privilege Statement [page 344]
REVOKE CHANGE PASSWORD Statement [page 333]
REVOKE SET USER Statement [page 342]

1.3.2.5 Users and Privileges Granted System Objects

Information about the current users of a database and their privileges is stored in the database system tables, which are accessible through system views.

Most system tables are owned by the SYS user ID. You cannot log in using the SYS user ID.

The DBA has SELECT access to all system tables, just as to any other tables in the database. The access of other users to some of the tables is limited. For example, only the DBA has access to the `SYS.SYSUSERPERM` table, which contains all information about the privileges of users of the database, as well as the passwords of each user ID. However, `SYS.SYSUSERPERMS` is a view that contains all information in `SYS.SYSUSERPERM` except passwords, and by default, all users have SELECT access to this view. All privileges and role memberships that are automatically set up in a new database for SYS and PUBLIC system roles, and DBA user can be fully modified.

In this section:

- User ID, Role, and Privilege Information in System Tables [page 104]
  System tables containing information about user IDs, roles, and privileges.
- User ID, Role, and Privilege Information in System Views [page 105]
  System views containing information about user IDs, roles, and privileges.

1.3.2.5.1 User ID, Role, and Privilege Information in System Tables

System tables containing information about user IDs, roles, and privileges.

All tables and views are owned by the SYS role, and their qualified names are SYS.ISYSUSERPERM, SYS.ISYSTABLEPERM, and so on. Execute the appropriate SELECT queries on these tables to generate all the user ID and privilege information stored in the database.

Table	Default	Contents
ISYSUSERPERM	SELECT ANY TABLE system privilege	Database-level privileges and password for each user ID
ISYSTABLEPERM	PUBLIC	All privileges on table given by the GRANT commands
ISYSCOLPERM	PUBLIC	All columns with UPDATE privilege given by the GRANT command
ISYSPROCPERM	PUBLIC	Each row holds one user who is granted the privilege to use one procedure

1.3.2.5.2 User ID, Role, and Privilege Information in System Views

System views containing information about user IDs, roles, and privileges.

In addition to this list, there are tables and views containing information about each object in the database.

View	Default	Contents
SYSUSERAUTH (deprecated)	SELECT ANY TABLE system privilege	All information in SYSUSERPERM (deprecated) except user numbers
SYSUSERPERMS (deprecated)	PUBLIC	All information in SYSUSERPERM (deprecated) except passwords
SYSUSERLIST (deprecated)	PUBLIC	All information in SYSUSERAUTH (deprecated) except passwords
SYSTABAUTH	PUBLIC	Information from SYSTABLEPERM in a more readable format
SYSCOLAUTH	PUBLIC	Information from SYSCOLPERM in a more readable format
SYSPROCAUTH	PUBLIC	Information from SYSPROCPERM in a more readable format

1.3.2.6 Stored Procedure to Map System Privileges to System Roles

The `sp_sys_priv_role_info` stored procedure generates a report that maps each system privilege role to a system role.

A separate row is generated for each system privilege. No system privileges are required to execute the procedure.

1.3.3 Object-Level Privileges

Database object-level privileges can be granted to and revoked from users.

In this section:

Ownership Privileges of Database Objects [page 106]
Ownership of a database object carries with it privileges to carry out actions on that object.

Inheritance of Database Privileges [page 106]
You can grant database privileges directly to users, or they can be inherited through role membership.

Grant and Revoke Object-Level Privileges [page 107]
You can grant to users, or revoke from them, combinations of privileges to define their access to database objects.

Privileges Required to Manage Table Objects in a Dbspace [page 119]
The privileges required depend on the task you are performing.

Command Line Options That Control Privileges [page 120]
The database server start-up command `start_iq` includes options that set the privilege level of some database and server functions.

Revoking the Privilege to Run a Procedure [page 121]
Remove the privilege to execute or call a specific procedure.

Stored Procedure to Display Object-Level Privileges Granted [page 122]
Execute the `sp_objectpermission` stored procedure to generate a report on object-level privileges granted to the specified role or user name or object privileges granted on the specified object or dbspace.

1.3.3.1 Ownership Privileges of Database Objects

Ownership of a database object carries with it privileges to carry out actions on that object.

The creator of a database object may not necessarily be its owner. Another user can be designated as owner during the create process. If no owner is specified, the creator is the owner.

The owner of a table can modify the table structure, for instance, or can grant privileges to other database users to update the information within the table.

> i Note
>
> The owner of a table can load data if he or she has sufficient privilege, or if the server was started with the –`gl all` switch on the command line or configuration file. Ownership or the CREATE ANY OBJECT system privilege are insufficient to issue the `LOAD TABLE` command; the INSERT privilege on the table is also required.

A user with the ALTER ANY OBJECT system privilege can modify any database object (regardless of owner) that can be created using the CREATE ANY OBJECT system privilege. A user with the CREATE ANY OBJECT system privilege can create database objects to be owned by other users.

1.3.3.2 Inheritance of Database Privileges

You can grant database privileges directly to users, or they can be inherited through role membership.

Privilege Name	Supported By Database Object	Allows a User To
ALL	Tables, views, materialized views	Perform all tasks associated with tables, views and materialized views.
ALTER	Tables	Alter the structure of a table.

Privilege Name	Supported By Database Object	Allows a User To
CREATE	Dbspaces	Create objects on the dbspace. The additional privileges required depend on the object that is being created. For example, to create a table, one of CREATE TABLE, CREATE ANY TABLE, or CREATE ANT OBJECT is required.
DELETE	Tables, view	Delete rows from the table or view.
EXECUTE	Procedure, user-defined functions	Execute the procedure or user-defined function.
INSERT	Table, views	Insert rows into the table or view.
LOAD	Tables	Load the table if the `-gl` database option is set to anything other than NONE.
REFERENCES	Tables	Create indexes on a table, and to create foreign keys that reference a table.
SELECT	Table, views	Look at information in a table or view.
TRUNCATE	Table, materialized views	Truncate the table or materialized view.
UPDATE	Tables, views	Update rows in a table or view.
USAGE	Sequence generators	Evaluate the current or next value in the sequence.

In a multiplex, only write servers can modify table privileges on tables owned by the write server.

1.3.3.3 Grant and Revoke Object-Level Privileges

You can grant to users, or revoke from them, combinations of privileges to define their access to database objects.

In this section:

Grant the privilege to truncate a specified table.

Granting the UPDATE Privilege on Tables and Views [page 114]
Grant the privilege to modify the data in a table or view. This privilege can be restricted to a set of columns in the table.

Granting the Right to Administer an Object-Level Privilege [page 115]
Grant the privilege to allow a user to pass a specific object privilege on to other users.

Granting the CREATE Privilege on Dbspaces [page 116]
Grant the privilege to create database objects in the specified dbspace.

Granting the EXECUTE Privilege on Functions and Procedures [page 116]
Grant the privilege to run a procedure or user-defined function.

Granting the USAGE Privilege on Sequence Generators [page 117]
Grant the privilege to evaluate the current or next value in a sequence.

Revoking an Object-Level Privilege [page 118]
Remove the ability of a user to use a specific object-level privilege, or to grant the privilege to other users.

1.3.3.3.1 Granting the ALTER Privilege on Tables

Grant the privilege to alter the structure of a table. This privilege does not apply to views.

Prerequisites

Requires one of:

- The MANAGE ANY OBJECT PRIVILEGE system privilege or,
- The ALTER object privilege on the table with the WITH GRANT OPTION clause, or
- You own the table.

Procedure

To grant the ALTER privilege, enter:

```
GRANT ALTER
  ON <table_name>
  TO <userID [,...]>
```

Related Information

GRANT Object-Level Privilege Statement [page 321]

Granting the Right to Administer an Object-Level Privilege [page 115]

1.3.3.3.2 Granting the DELETE Privilege on Tables and Views

Grant the privilege to delete all data in a specified table or view.

Prerequisites

Requires one of:

- The MANAGE ANY OBJECT PRIVILEGE system privilege or,
- The DELETE object privilege on the table with the WITH GRANT OPTION clause, or
- You own the table.

Procedure

To grant the DELETE privilege, enter:

```
GRANT DELETE
  ON <table_name>
  TO <userID [,...]>
```

Related Information

GRANT Object-Level Privilege Statement [page 321]

Granting the Right to Administer an Object-Level Privilege [page 115]

1.3.3.3.3 Granting the INSERT Privilege on Tables and Views

Grant the privilege to insert data into a table or view.

Prerequisites

Requires one of:

- The MANAGE ANY OBJECT PRIVILEGE system privilege or,
- The INSERT object privilege on the table with the WITH GRANT OPTION clause or,
- You own the table.

Procedure

To grant the INSERT privilege, enter:

```
GRANT INSERT
  ON <table_name>
  TO <userID [,...]>
```

Related Information

GRANT Object-Level Privilege Statement [page 321]
Granting the Right to Administer an Object-Level Privilege [page 115]

1.3.3.3.4 Granting the LOAD Privilege on Tables

Grant the privilege to load a specified table.

Prerequisites

Requires one of:

- MANAGE ANY OBJECT PRIVILEGE system privilege or,
- The LOAD object privilege with the WITH GRANT OPTION clause on the table or,
- You own the table.

Procedure

To grant the LOAD privilege, enter:

```
GRANT LOAD
  ON <table_name>
  TO <userID> [,...]
```

Related Information

[GRANT Object-Level Privilege Statement [page 321]]

[Granting the Right to Administer an Object-Level Privilege [page 115]]

1.3.3.3.5 Granting the REFERENCES Privilege on Tables

Grant the privilege to indexes and to foreign keys on a table. This privilege does not apply to views. This privilege can be restricted to a set of columns in the table.

Prerequisites

Requires one of:

- The MANAGE ANY OBJECT PRIVILEGE system privilege or,
- The REFERENCES object privilege on the table with the WITH GRANT OPTION clause or,
- You own the table.

Procedure

To grant the REFERENCES privilege, enter:

```
GRANT REFERENCES <column_name>
  ON <table_name>
  TO <userID [,...]>
```

Example

This statement grants the REFERENCES privilege to user Joe on columns Col_1 and Col_2 in the table named sales_table:

```
GRANT REFERENCES Col_1, Col_2 ON sales_table
TO Joe
```

Related Information

GRANT Object-Level Privilege Statement [page 321]
Granting the Right to Administer an Object-Level Privilege [page 115]

1.3.3.3.6 Granting the SELECT Privilege on Tables and Views

Grant the privilege to select data in a table or view, but not to alter it. This privilege can be restricted to a set of columns in the table.

Prerequisites

Requires one of:

- The MANAGE ANY OBJECT PRIVILEGE system privilege or,
- The SELECT object privilege on the table with the WITH GRANT OPTION clause or,
- You own the table.

Procedure

To grant the SELECT privilege, enter:

```
GRANT SELECT <column_name>
  ON <table_name>
  TO <userID [,...]>
```

Example

This statement grants the SELECT privilege to user `Joe` on columns `Col_1` and `Col_2` in the table named `sales_table`:

```
GRANT SELECT Col_1, Col_2 ON sales_table
TO Joe
```

Related Information

GRANT Object-Level Privilege Statement [page 321]
Granting the Right to Administer an Object-Level Privilege [page 115]

1.3.3.3.7 Granting the TRUNCATE Privilege on Tables

Grant the privilege to truncate a specified table.

Prerequisites

Requires one of:

- The MANAGE ANY OBJECT PRIVILEGE system privilege or,
- The TRUNCATE object privilege with the WITH GRANT OPTION clause on the table or,
- You own the table.

Procedure

To grant the TRUNCATE privilege, enter:

```
GRANT TRUNCATE
  ON <table_name>
  TO <userID> [,...]
```

Related Information

GRANT Object-Level Privilege Statement [page 321]
Granting the Right to Administer an Object-Level Privilege [page 115]

1.3.3.3.8 Granting the UPDATE Privilege on Tables and Views

Grant the privilege to modify the data in a table or view. This privilege can be restricted to a set of columns in the table.

Prerequisites

Requires one of:

- The MANAGE ANY OBJECT PRIVILEGE system privilege or,
- The UPDATE object privilege on the table with the WITH GRANT OPTION clause or,
- You own the table.

Procedure

To grant the UPDATE privilege, enter:

```
GRANT UPDATE <column_name>
  ON <table_name>
  TO <userID [,...]>
```

Example

This statement grants the UPDATE privilege to user `Joe` on columns `Col_1` and `Col_2` in the table named `sales_table`:

```
GRANT UPDATE Col_1, Col_2 ON sales_table
TO Joe
```

Related Information

GRANT Object-Level Privilege Statement [page 321]

Granting the Right to Administer an Object-Level Privilege [page 115]

1.3.3.3.9 Granting the Right to Administer an Object-Level Privilege

Grant the privilege to allow a user to pass a specific object privilege on to other users.

Prerequisites

At least one of these conditions:

- You created the table.
- Privileges on the table with the ADMIN OPTION.
- LOAD and TRUNCATE object privileges.
- The MANAGE ANY OBJECT PRIVILEGE system privilege. If the LOAD or TRUNCATE object privilege is granted using the WITH GRANT OPTION clause, the grantee can then grant the object privilege to other users, but is limited to those tables specified in the original GRANT statement. Under this scenario, the grantee does not need the MANAGE ANY OBJECT PRIVILEGE system privilege.

Procedure

1. Connect to the database.
2. To grant the right to grant a privilege to another user, enter:

```
GRANT <Object_privilege _name>
  ON <table_name>
  TO <userID [,...]>
  WITH GRANT OPTION
```

Example

This statement grants the privilege to `Mary` to perform deletions on the table `Sales`:

```
GRANT DELETE ON Sales TO Mary
```

This statement grants the right to `Joe` to both perform deletions on the table `Sales`, and to grant the DELETE privilege to other users:

```
GRANT DELETE ON Sales TO Joe
WITH GRANT OPTION
```

Related Information

GRANT Object-Level Privilege Statement [page 321]
Granting the Right to Administer an Object-Level Privilege [page 115]

1.3.3.3.10 Granting the CREATE Privilege on Dbspaces

Grant the privilege to create database objects in the specified dbspace.

Prerequisites

Requires the MANAGE ANY DBSPACE system privilege.

Procedure

To grant the CREATE privilege, enter:

```
GRANT CREATE
  ON <dbspace_name>
  TO <userID> [,...]
```

Related Information

GRANT CREATE Statement [page 320]

1.3.3.3.11 Granting the EXECUTE Privilege on Functions and Procedures

Grant the privilege to run a procedure or user-defined function.

Prerequisites

Requires one of:

- The MANAGE ANY OBJECT PRIVILEGE system privilege or,
- You own the procedure.

Procedure

To grant the EXECUTE privilege, enter:

```
GRANT EXECUTE
  ON <procedure_name>
  TO <userID> [,...]
```

Related Information

GRANT EXECUTE Statement [page 321]

1.3.3.3.12 Granting the USAGE Privilege on Sequence Generators

Grant the privilege to evaluate the current or next value in a sequence.

Prerequisites

Requires one of:

- The MANAGE ANY OBJECT PRIVILEGE system privilege or,
- You own the sequence generator.

Procedure

To grant the USAGE privilege, enter:

```
GRANT USAGE
  ON <sequence_name>
  TO <userID> [,...]
```

Related Information

GRANT USAGE ON SEQUENCE Statement [page 333]

1.3.3.3.13 Revoking an Object-Level Privilege

Remove the ability of a user to use a specific object-level privilege, or to grant the privilege to other users.

Prerequisites

Grantor must have at least one of these conditions:

- Be the original grantor of the privilege that is being revoked or,
- Have the MANAGE ANY OBJECT PRIVILEGE system privilege.

Context

If you revoke a privilege from a user who has been granted a privilege with the WITH GRANT OPTION clause, then everyone to whom that user granted the privilege also has his or her privilege revoked. For example, you granted `User1` the SELECT privilege with the WITH GRANT OPTION clause. `User1` then grants the SELECT privilege to `User2`. If you revoke the SELECT privilege from `User1`, it is also revoked from `User2`.

The `REVOKE` command applies to the object-level privilege itself, not to any administrative right granted on the privilege. Therefore, you cannot revoke administrative rights only and leave the object-level privilege intact. To correctly remove a user's administrative rights only to an object-level privilege, you must first revoke the privilege and then regrant the privilege without the WITH GRANT OPTION clause.

Procedure

1. To revoke an object-level privilege, including any administrative privilege, execute:

```
REVOKE <object_privilege_name>
  ON <table_name>
  FROM <userID [,...]>
```

2. (Optional) To then regrant the object-level privilege without administrative rights, execute:

```
GRANT <object_privilege_name>
  ON <table_name>
  TO <userID [,...]>
```

Example

This example assumes that Joe has been granted the right to both perform deletions on the Sales table, and to grant the DELETE object-level privilege on the table to other users.

This statement revokes all DELETE object-level privileges on the table Sales, which by definition includes any administrative rights:

```
REVOKE DELETE ON Sales FROM Joe
```

This statement regrants the object-level privilege only, with no administrative rights:

```
GRANT DELETE ON Sales TO Joe
```

Related Information

REVOKE Object-Level Privilege Statement [page 338]
REVOKE CREATE Statement [page 336]
REVOKE EXECUTE Statement [page 337]
REVOKE USAGE ON SEQUENCE Statement [page 347]

1.3.3.4 Privileges Required to Manage Table Objects in a Dbspace

The privileges required depend on the task you are performing.

To create a new table on a dbspace requires the CREATE object-level privilege on the dbspace. To move an existing table or column to a dbspace requires the MANAGE ANY DBSPACE system privilege or the CREATE object-level privilege on the destination dbspace.

In addition to the dbspace requirements, you also require a system privilege for the specific task. For example, you need the CREATE TABLE or CREATE ANY TABLE system privilege to create a table, the ALTER ANY TABLE system privilege to alter the table, and so on.

For example, to create table1, owned by you, in dbspace test1, you require the CREATE object-level privilege on test1, as well as the CREATE TABLE system privilege. To then move table1 from dbspace test1 to dbspace test2 requires either the MANAGE ANY DBSPACE system privilege or the CREATE object-level privilege on test2, the destination dbspace.

You can grant the required privileges to, or revoked them from, a user or a role. Any member in a role inherits the privileges from the role.

By default, the CREATE object-level privilege on IQ_SYSTEM_MAIN, IQ_SYSTEM_TEMP, and SYSTEM is granted to PUBLIC.

1.3.3.5 Command Line Options That Control Privileges

The database server start-up command `start_iq` includes options that set the privilege level of some database and server functions.

Switches That Start and Stop Databases

The `-gd` option lets you limit the users who can start or stop a database on a running server to those with a certain level of privilege in the database to which they are already connected:

- `DBA` – (default value) only users with SERVER OPERATOR system privilege can start an extra database.
- `ALL` – (default in `start_iq` and `default.cfg`) any user can start and stop databases. This setting means that the DBA does not need to issue `START DATABASE` commands. Users must still be granted the privileges to access a particular database once he or she has started it.
- `NONE` – no one can start or stop a database from Interactive SQL on a running server.

> i Note
>
> If `-gd ALL` is not set when you start the server, only a user with the SERVER OPERATOR system privilege can start additional databases on that server. This means that users cannot connect to databases that are not already started, either at the same time as the server, or since then by a user with the SERVER OPERATOR system privilege. However, it also lets a user without the SERVER OPERATOR system privilege stop a database. For this reason, you may want to change this setting to DBA on production databases.

Switches That Create and Delete Databases

The `-gu` option limits the users who can create and drop databases to those with a certain level of privilege in the database to which they are connected.

- `DBA` – only users with SERVER OPERATOR system privilege can create and drop databases.
- `ALL` (default) – any user can create and drop databases.
- `NONE` – no user can create or drop a database.
- `UTILITY_DB` – only those users who can connect to the `utility_db` database can create and drop databases.

Stop Server Switch

The `-gk` option limits the users who can shut down a server with the `dbstop` utility or `STOP ENGINE` command:

- `DBA` (default) – only users with SERVER OPERATOR system privilege can stop the server.
- `ALL` – any user can stop the server.

- `NONE` – no user can shut down the server with the `dbstop` utility or `STOP ENGINE` command.

Switches That Load and Unload Databases

The `-gl` option limits the users who can load data using `LOAD TABLE` to users with a certain level of privilege in the database.

- `DBA` – any user with the LOAD ANY TABLE, ALTER ANY TABLE, or ALTER ANY OBJECT system privilege can load data.
- `ALL` (default for `start_iq` and `default.cfg`) – any user can load data.
- `NONE` – data cannot be loaded.

Related Information

-gl database server option [page 364]
-gk database server option [page 363]
-gu database server option [page 364]

1.3.3.6 Revoking the Privilege to Run a Procedure

Remove the privilege to execute or call a specific procedure.

Prerequisites

Revoker must either:

- Be the original grantor of the privilege that is being revoked or,
- Have the MANAGE ANY OBJECT PRIVILEGE system privilege.

Procedure

To revoke the EXECUTE privilege to run a specific procedure, execute:

```
REVOKE EXECUTE ON <procedure_name>
FROM <grantee [,...]>
```

Related Information

REVOKE EXECUTE Statement [page 337]

1.3.3.7 Stored Procedure to Display Object-Level Privileges Granted

Execute the `sp_objectpermission` stored procedure to generate a report on object-level privileges granted to the specified role or user name or object privileges granted on the specified object or dbspace.

The report includes the user ID of the privilege grantor and grantee, the object name and owner, the privilege granted, and whether the grantee can in turn grant the privilege to other users.

No system privileges are required to execute the procedure on your user ID. To execute `sp_objectpermission` on other users or a dbspace, you must have MANAGE ANY OBJECT PRIVILEGE or MANAGE ANY DBSPACE privilege, respectively.

Related Information

sp_objectpermission System Procedure [page 439]

1.3.4 System Procedure Privileges

There are two security models under which privileged system procedures can run. Each model grants the ability to run the system procedure differently.

> i Note
>
> The following information applies only to SAP IQ privileged system procedures, not user-defined stored procedures.

The first model, called the SYSTEM PROCEDURE DEFINER model, runs a privileged system procedure with the privileges of its owner, typically dbo. The second model, called the SYSTEM PROCEDURE INVOKER model, runs a privileged system procedure with the privileges of the person executing it.

To run a privileged system procedure using the SYSTEM PROCEDURE DEFINER model, grant explicit EXECUTE object-level privilege on the procedure. Any system privileges required to run any underlying authorized tasks of the system procedure are automatically inherited from the owner (definer of the system procedure).

For privileged system procedures using the SYSTEM PROCEDURE INVOKER model, the EXECUTE object-level privilege is granted to the PUBLIC role, and since, by default, every user is a member of the PUBLIC role, every user automatically inherits the EXECUTE object-level privilege. However, since the PUBLIC role is not the owner of the system procedures, and is not granted any system privileges, the system privileges required to run any underlying authorized tasks must be granted directly or indirectly to the user.

By default, a database created in versions 16.0 and later runs all privileged system procedures using the SYSTEM PROCEDURE INVOKER model. A database created in versions earlier than 16.0 and upgraded to versions 16.0 and later runs privileged system procedures using a combination of both the SYSTEM PROCEDURE DEFINER and SYSTEM PROCEDURE INVOKER models. In the combined model, all pre-16.0 privileged system procedures use the SYSTEM PROCEDURE DEFINER model, and any privileged system procedures introduced with 16.0 (or any future release) use the SYSTEM PROCEDURE INVOKER model. You can override the default security model when creating or upgrading a database, or any time thereafter. However, SAP recommends that you not do so, as it may result in loss of functionality on custom stored procedures and applications.

In this section:

1.3.4.1 Granting the Ability to Run a Privileged System Procedure

The process by which you grant the ability to run a privileged system procedure is dependent on the security model under which it runs.

For a privileged system procedure using the SYSTEM PROCEDURE DEFINER model, grant EXECUTE object-level privilege on the system procedure to the user:

```
GRANT EXECUTE ON <sys_procedure_name>
    TO <grante> [, …]
```

For a privileged system procedure using the SYSTEM PROCEDURE INVOKER model, grant the underlying system privileges required by the system procedure to the user. Use `sp_proc_priv()` to identify the system privileges required to run a system procedure.

```
GRANT <system_privilege_name>
    TO <grantee> [, …]
```

Related Information

GRANT EXECUTE Statement [page 321]

1.3.4.2 Revoking the Ability to Run a Privileged System Procedure

The process by which you revoke the ability to run a privileged system procedure is dependent on the security model under which it runs.

For a privileged system procedure using the SYSTEM PROCEDURE DEFINER model, revoke the EXECUTE object-level privilege on the system procedure from the user:

```
REVOKE EXECUTE ON <sys_procedure_name>
    FROM <grantee> [, …]
```

For a privileged system procedure using the SYSTEM PROCEDURE INVOKER model, revoke the underlying system privileges required by the system procedure from the user:

```
REVOKE <system_privilege_name>
    FROM <grantee> [, …]
```

Related Information

REVOKE EXECUTE Statement [page 337]

1.3.4.3 Determining the Security Model Used by a Database

There are two security models a database can use.

To determine the security model a database is using, execute:

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

1 indicates the database is using the SYSTEM PROCEDURE INVOKER model. 0 indicates that the database is using the combined model.

In the combined model, only pre-16.0 privileged system procedures run using the SYSTEM PROCEDURE DEFINER. Refer to the pre-16.0 privileged system procedures list to identify these system procedures.

A new or upgraded 16.0 or later database cannot be configured to run all system procedures using the SYSTEM PROCEDURE DEFINER model.

1.3.4.4 Pre-16.x Privileged System Procedures

A list of pre-16.x 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 *SAP IQ 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.x 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 *SAP IQ 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

1.4 Passwords

A user can be granted the ability to manage other users' passwords. You can configure password management to require one or two users to complete a password change.

In this section:

1.4.1 Passwords in the Database

As of version 15.0, SAP IQ uses SHA256 to hash passwords. Passwords are stored in UTF-8.

When passwords are created or changed, they are converted to UTF-8 before being hashed and stored in the database. If the database is unloaded and reloaded into a database with a different character set, existing passwords continue to work. If the server cannot convert from the client's character set to UTF-8, SAP then recommends that passwords be composed of 7-bit ASCII characters as other characters may not work correctly.

1.4.2 Granting the CHANGE PASSWORD System Privilege to a User

Allow a user to manage the password of other users.

Prerequisites

- The CHANGE PASSWORD system privilege granted with administrative rights.
- Each target user specified (target_users_list) is an existing user or user-extended role with a login password.
- Each target role specified (target_roles_list) must be an existing user-extended or user-defined role.

Context

You can grant a user the ability to change the password of any user in the database (ANY), only specific users (`<target_users_list>`), or members of specific roles (ANY WITH ROLES `<target_roles_list>`). Administrative rights to the CHANGE PASSWORD system privilege can be granted only when using the ANY clause.

If no clause is specified, the default is ANY, WITH NO ADMIN OPTION.

When regranting the CHANGE PASSWORD system privilege, the effect of the grant is cumulative. For example, if you grant `User1` the privilege limited to `User2` and `User3`, and then regrant the privilege limited to `Role1`, `User1` can manage the password of `User2`, `User3`, and any member of `Role1`.

If you grant the CHANGE PASSWORD system privilege to a user with fewer rights than currently granted, the higher rights are retained. For example, if the privilege is granted using the ANY clause and then regranted using the `<target_users_list>` clause, the user retains the rights of the ANY clause.

Procedure

To grant the CHANGE PASSWORD system privilege, execute one of these statements:

Grant Type	Statement
Any database user, with	`GRANT CHANGE PASSWORD(ANY)`

Grant Type	Statement
full administrative rights	TO <user_ID> WITH ADMIN OPTION
Any database user, with administrative rights only	GRANT CHANGE PASSWORD(ANY) TO <user_ID> WITH ADMIN ONLY OPTION
Any database user, with no administrative rights	GRANT CHANGE PASSWORD(ANY) TO <user_ID> WITH NO ADMIN OPTION
Specified users, with no administrative rights	GRANT CHANGE PASSWORD(<target_users_list>) TO <user_ID> WITH NO ADMIN OPTION
Any member of specified roles, with no administrative rights	GRANT CHANGE PASSWORD(ANY WITH ROLES <target_roles_list>) TO <user_ID> WITH NO ADMIN OPTION
Specified users, or any member of specified roles, with no administrative rights	GRANT CHANGE PASSWORD (<target_users_list>),(ANY WITH ROLES <target_roles_list>) TO <user_ID> WITH NO ADMIN OPTION

Example

This statement grants `Sam` the ability to change the password of any database user:

```
GRANT CHANGE PASSWORD (ANY) TO Sam
or
GRANT CHANGE PASSWORD TO Sam
```

This statement grants `Sally` and `Bob` the ability to change the password for `Jane`, `<Joe>`, and `Laurel` only:

```
GRANT CHANGE PASSWORD (Jane, Joe, Laurel) TO Sally, Bob
```

This statement grants `Mary` the ability to change the password of any member of the `Sales1` role:

```
GRANT CHANGE PASSWORD (ANY WITH ROLES Sales1) TO Mary
```

This statement grants Sarah the ability to change the password of Joe or Sue, or any member of the Sales2 role:

```
GRANT CHANGE PASSWORD (Joe, Sue), (ANY WITH ROLES Sales2) TO Sarah
```

This statement grants Joan the ability to change the password of any member of the Marketing1 or Marketing2 roles:

```
GRANT CHANGE PASSWORD (ANY WITH ROLES Marketing1, Marketing2) TO Joan
```

Related Information

GRANT CHANGE PASSWORD Statement [page 316]

1.4.3 Revoking the CHANGE PASSWORD System Privilege from a User

Remove the ability of a user to manage passwords and administer the system privilege.

Prerequisites

Requires the CHANGE PASSWORD system privilege granted with administrative rights.

Context

You can grant the CHANGE PASSWORD system privilege to a user multiple times, using different clauses. For example, User1 is granted the CHANGE PASSWORD system privilege once using the ANY clause and again with the <target_users_list> clause. In cases of multiple grants, the same form of the clause used for the GRANT statement must be used to revoke it.

Continuing with the example, if the system privilege is revoked from User1 using the ANY clause, the grant with the <target_users_list> clause remains in effect. The net effect is that User1 is now limited to managing the passwords of users on the <target_users_list>. Alternately, if the system privilege is revoked from User1 using the <target_users_list> clause, the grant with the ANY clause remains in effect. The net effect in this scenario is that User1 can continue to manage the passwords of any user in the database.

Procedure

To revoke the CHANGE PASSWORD system privilege, execute one of these statements:

Revoke Type	Description

Administrative rights to system privilege only	`REVOKE ADMIN OPTION FOR CHANGE PASSWORD(ANY)`
	`FROM<user_ID [,...]>`

System privilege to manage password of any database user, including administrative rights	`REVOKE CHANGE PASSWORD`
	`FROM<user_ID [,...]>`

System privilege to manage password of specified users	`REVOKE CHANGE PASSWORD(<target_users_list>)`
	`FROM<user_ID [,...]>`

System privilege to manage password of specified roles	`REVOKE CHANGE PASSWORD(ANY WITH ROLES <target_roles_list>)`
	`FROM<user_ID [,...]>`

Example

Both these statements remove the ability of `Sam` to change the password of any database user:

```
REVOKE CHANGE PASSWORD (ANY) FROM Sam
or
GRANT CHANGE PASSWORD TO Sam
```

Assuming that `Frank` was granted the CHANGE PASSWORD system privilege with the ANY and WITH ADMIN OPTION clauses, this statement removes only the ability to administer the system privilege from `Frank`. He can continue to change the password of any user in the database.

```
REVOKE ADMIN OPTION FOR CHANGE PASSWORD (ANY) FROM Frank
```

This statement removes the ability of `Sally` and `Bob` to change the password of `Jane`, `Joe`, and `Laurel` only:

```
REVOKE CHANGE PASSWORD (Jane, Joe, Laurel) FROM Sally, Bob
```

This statement removes the ability of `Mary` to change the password of any member of the `Sales1` role:

```
REVOKE CHANGE PASSWORD (ANY WITH ROLES Sales1) FROM Mary
```

This statement removes the ability of `Sarah` to change the password of `Joe` or `Sue`, or any member of the `Sales2` role:

```
REVOKE CHANGE PASSWORD (Joe, Sue), (ANY WITH ROLES Sales2) FROM Sarah
```

This statement removes the ability of `Joan` to change the password of any member of the `Marketing1` or `Marketing2` roles:

```
REVOKE CHANGE PASSWORD (ANY WITH ROLES Marketing1, Markeing2) FROM Joan
```

Related Information

REVOKE CHANGE PASSWORD Statement [page 333]

1.4.4 Changing a Password – Single Control

A single user can manage the password of another user.

Prerequisites

- The CHANGE PASSWORD system privilege.
- The managing user has been granted the right to change the password of the target user.

Procedure

At a command prompt, type:

```
ALTER USER <userID>
   IDENTIFIED BY <password>
```

Related Information

Case-Sensitivity of User IDs and Passwords [page 149]
ALTER USER Statement [page 294]

1.4.5 Dual Control Password Management Option

The Dual Control Password option requires two administrative users to change the password of a target user, thus ensuring that no single user knows (or controls) the password of the target user.

Two distinct administrative users are required to generate each part of the new password. It is the combination of the two parts that become the new password for the target user. The same user cannot generate both password parts. If the same user attempts to define both password parts, the server displays an error message, and the second password part is not set.

If the server is restarted after the first password part is specified, but before the second password part is specified, the first password part is not lost. When the second password part is specified by a different user, the dual password change process completes successfully. The target user can then log in using the combined password parts.

Once initiated, generation of the dual passwords for the target user can be cancelled by specifying "NULL" as the password, as long as the user has been granted the CHANGE PASSWORD system privilege, and the right to manage the password of the target user.

Each administrative user setting a password part must notify the target user of the new password part, and indicate whether it is the first or second part. To use the password, the target user enters the dual password in first part, second part order. There is a 127-character limit for each part.

If the target user is not logged in when the dual password change process completes, he or she simply logs in. Once the dual password is accepted, the user is immediately prompted to change his or her password. This provides the final level of password security. If the user is already logged in when the dual password change process completes, the user can use the `ALTER USER` or `GRANT CONNECT` statements, or the `sp_password` or `sp_iqpassword` system procedures to change the password. At the prompt for the current password, enter the new dual part passwords, not the password originally entered for the current session.

The Change Password Dual Control option is enabled in a login policy.

In this section:

Related Information

Case-Sensitivity of User IDs and Passwords [page 149]
ALTER USER Statement [page 294]
GRANT CONNECT Statement [page 318]
sp_iqpassword Procedure [page 437]

1.4.5.1 Enabling Dual Control for Changing Passwords

Require input from two administration users to change the password of another user.

Prerequisites

The MANAGE ANY LOGIN POLICY OPTION system privilege.

Context

Dual control for managing passwords is a configurable option in a login policy. By default, it is disabled (OFF).

Procedure

To enable the option, execute:

```
ALTER LOGIN POLICY <policy-name>
CHANGE_PASSWORD_DUAL_CONTROL=ON
```

Related Information

ALTER LOGIN POLICY Statement [page 284]
CREATE LOGIN POLICY Statement [page 300]

1.4.6 Changing a Password – Dual Control

Two users are required to manage the password of another user.

Prerequisites

- The CHANGE PASSWORD system privilege.
- The managing user has been granted the right to change the password of the target user.
- The CHANGE_PASSWORD_DUAL_CONTROL option is enabled in the login policy of the managing user.

Procedure

1. At a command prompt, the first managing user enters:

   ```
   ALTER USER <userID>
   IDENTIFIED FIRST BY <password_part1>
   ```

2. At a command prompt, the second managing user enters:

   ```
   ALTER USER <userID>
   IDENTIFIED LAST BY <password_part1>
   ```

Example

Assuming login policy Sales1 has the CHANGE_PASSWORD_DUAL_CONTROL option enabled, User3 is assigned Sales1, and User1 and User2 have been granted the necessary privileges to change the password of User3, these statements set the two password parts for User3 to <NewPassPart1> and <NewPassPart2>:

User1 types:

```
ALTER USER user3 IDENTIFIED FIRST BY NewPassPart1
```

User2 types:

```
ALTER USER user3 IDENTIFIED LAST BY NewPassPart2
```

Related Information

Case-Sensitivity of User IDs and Passwords [page 149]
ALTER USER Statement [page 294]

1.5 Impersonation

A user can temporarily assume (impersonate) the specific roles and system privileges of another user to perform operations, provided he or she already has the minimum required privileges to perform the task to begin with.

Suppose User1 is responsible for performing a key task, but he or she is unavailable. User2 has sufficient privileges to complete the task, but has additional privileges not available to User1. If User2 performs the task, it could complete differently than when performed by User1. To avoid this, User2 temporarily assumes (impersonates) the roles and system privileges specific to User1, and performs the task.

Impersonation is achieved by first granting a user the SET USER system privilege, and then issuing the SETUSER statement to initiate the impersonation.

> i Note
>
> The SET USER system privilege is two words; the SETUSER statement is one word.

When you grant the SET USER system privilege, you can define the scope of impersonation as:

- Any user in the database.
- Any user within a specified list of users (<target_users_list>).
- Any user who is a member of one or more of the specified roles (<target_roles_list>).

To impersonate another user, the impersonating (grantee) user must have been granted, at minimum, all of the roles and system privileges, with the same or higher administrative privileges, as the impersonated (target) user. This is called the at-least criteria. The impersonating user can have been granted additional roles, system privileges, or higher administrative privileges, but not fewer. While impersonating another user, you can grant additional roles and privileges to, or revoke from the impersonator or impersonate as long as doing so does not violate the at-least criteria. If the grant or revoke violates the criteria, an error message appears, and the statement fails.

For example, `User1` is successfully impersonating `User2`. You grant a new role to `User1`, but not to `User2`. Since this grant does not cause a violation of the criteria for `User1` to impersonate `User2` (`User1` still has at least the same roles and privileges granted to `User2`), the grant is successful. If, however, new role is granted to `User2` instead of `User1`, the grant statement fails because it results in `User2` being granted more roles than `User1`.

When you impersonates another user, the user ID of the impersonated user (not yours), appears in the audit logs. However, since the act of impersonation (issuance of the SETUSER command) is also recorded in the audit logs, you can determine whether the grantee or target user executed a task.

In a multiplex configuration, if an impersonation is active in a connection that is present in the coordinator, and an attempt is made to grant or revoke roles and privileges that violates the at-least criteria, the connection containing the active impersonation terminates. Since terminating the connection also terminates the impersonation, violation of at-least criteria is no longer an issue, and the `GRANT` or `REVOKE` statement executes successfully.

In this section:

1.5.1 Requirements for Impersonation

A user can successfully impersonate another user only if a specific set of criteria is met, also called the at-least requirements.

There are four criteria to successful impersonation:

1. The impersonator has been granted the right to impersonate the target user.
2. The impersonator has, at minimum, all the roles and system privileges granted to the target user.
3. The impersonator has been granted the said roles and system privileges with similar or higher administrative rights.

> **i Note**
>
> For the purposes of meeting administrative rights criteria, the WITH ADMIN OPTION and WITH ADMIN ONLY OPTION clauses are considered to grant similar administrative rights. They are also considered to grant higher administrative rights than the WITH NO ADMIN OPTION clause. For example, `User1` is granted `Role1` with the WITH ADMIN OPTION clause, `User2` is granted `Role1` with the WITH ADMIN ONLY clause, and `User3` is granted `Role1` with the WITH NO ADMIN OPTION clause. `User1` and `User2` are said to be granted `Role1` with similar administrative rights. `User1` and `User2` are also said to be granted `Role1` with higher administrative rights than `User3`.

4. If the target user has been granted a system privilege that supports extensions, the clauses used to grant the system privilege to the impersonator are a super-set of those used for the target user. Only the SET USER and CHANGE PASSWORD system privileges support extensions.
   - The ANY clause is considered a super-set of the `<target_roles_list>` and `<target_users_list>` clauses. If the target user has been granted the SET USER system privilege with an ANY grant, the impersonator must also have the ANY grant.
   - If the target user has been granted the SET USER system privilege with both the `<target_roles_list>` and `<target_users_list>` clauses, the impersonator must also have been granted the system privilege with the two clauses, and the target list of each clause must be equal to, or a super set of, the corresponding clause grant of the target user. For example, if the target lists of both the impersonator and target user contain `User1`, `User2` and `Role1`, `Role2`, respectively, the target list grants for each clause are said to be equal. Alternately, if the target list grants of the impersonator contain `User1`, `User2`, and `Role1`, `Role2`, respectively, while the target list grants of the target user contain `User1` and `Role2` only, the target list grants of the impersonator are said to be a super-set of the target user.
   - If the target user has been granted the SET USER system privilege with a single target list clause, the target list of the impersonator must be equal to or a super-set of the list of the target user. For example, the `<target_user_list>` of both the impersonator and the target user contain `User1` and `User2` (equal) or the impersonator list contains `User1`, `User2`, while the target user contains `User2`; `User1`, `User2` (impersonator list) is a super-set of `User2` (target user list).
   - By definition, a user can always impersonate himself or herself. Therefore, if the target user is granted the right to impersonate the impersonator, this does not violate the equal to or a super-set of criteria requirement of the impersonator. For example, `User3` is the impersonator and `User4` is the target user. The `<target_user_list>` for `User3` contains `User4` and `User5`. The `<target_user_list>` for `User4` contains `User3` and `User5`. If you remove the impersonator from the target list, the target list of `User3` meets the criteria requirement.

Scenario 1

Assuming that criteria 2 and 3 are met, consider the following scenario:

- There are five users: `<User1>`, `<User2>`, `<User3>`, `<User4>`, and `<User5>`.

- There are two roles: <Role1> and <Role2>.
- <User1> has been granted the SET USER system privilege with the ANY clause.
- <User2> has been granted the SET USER system privilege with the <target_users_list> clause for <User1> and <User4>.
- <User3> has been granted the SET USER system privilege with the <target_users_list> clause for <User1>, <User2>, <User4> and, <User5>, and the ANY WITH ROLES <target_roles_list> clause for <Role1> and <Role2>.
- <User4> has been granted the SET USER system privilege with the ANY clause and the <target_roles_list> clause for <Role1>.
- <User5> has been granted the SET USER system privilege with the <target_users_list> clause for <User4> and the ANY WITH ROLES <target_roles_list> for <Role1>.

<User1> and <User4> can successfully impersonate <User2>, <User3>, and <User5> because each is granted the SET USER system privilege with the ANY clause (criteria 4).

<User1> and <User4> can impersonate each other because they each have the ANY grant (criteria 4).

<User2>, <User3>, and <User5> cannot impersonate <User1> or <User4> because they do not have the ANY grant (criteria 4).

<User2> cannot impersonate <User3> or <User5> because:

- <User2> is not granted the right to impersonate these users (criteria 1).
- The SET USER system privilege is not granted to <User2> with the <target_roles_list> clause (criteria 4).

<User3> can successfully impersonate <User2> because:

- <User3> is granted the right to impersonate <User2> via the <target_users_list> clause (criteria 1).
- The <target_users_list> clause for <User3> is a super-set of <User2> (criteria 4). Though <User3> has a grant with the <target_role_list> clause, it is not required to satisfy the requirements for impersonation of <User2> because the latter does not have the same grant.

<User3> can successfully impersonate <User5> because:

- <User3> is granted the right to impersonate <User5> via the <target_users_list> clause (criteria 1).
- The <target_users_list> clause list for <User3> is a super-set of <User5> (criteria 4).
- The <target_roles_list> clause lists for <User3> and <User5> are equivalent (criteria 4).

<User5> cannot impersonate any other user because:

- <User1> and <User4> have an ANY grant (Criteria 4).
- <User2> and <User3> have a grant with a <target_users_list> clause that is not a sub-set of the grant to <User5> (criteria 4).
- <User3> has a grant with a <target_roles_list> clause that is not a subset (criteria 4).

Scenario 2

Assuming that criteria 1 and 4 are met, consider the following:

- There are two users: <User6> and <User7>.

- There are two roles: <Role4> and <Role5>.
- <User6> has been granted <Role4> with the WITH ADMIN OPTION clause, <Role5> with the WITH ADMIN ONLY OPTION clause, and the MANAGE ANY USER system privilege with the WITH ADMIN OPTION clause.
- <User7> has been granted <Role4> with the WITH ADMIN OPTION clause and <Role5> with the WITH NO ADMIN OPTION clause.

<User6> can successfully impersonate <User7> because:

- Both <User6> and <User7> are granted <Role4> and <Role5>. It does not matter that <User6> is granted additional privileges (MANAGE ANY USER system privilege) (criteria 2).
- <User6> is granted <Role4> with equivalent administrative rights as <User7>. <User6> is granted <Role5> with higher administrative rights than <User7> (criteria 3).

<User7> cannot impersonate <User6> because:

- <User7> is granted <Role4> and <Role5>, but not the MANAGE ANY USER system privilege (criteria 2).
- <User7> is granted <Role5> with lower administrative rights than <User6> (criteria 3).

Scenario 3

Consider the following:

- There are three users: <User8>, <User9> and <User10>.
- There are two roles: <Role5> and <Role6>.
- <User8> has been granted <Role5> with the WITH ADMIN OPTION clause, and the MANAGE ANY USER system privilege with the WITH ADMIN OPTION clause.
- <User9> and <User10> has been granted <Role5> with the WITH NO ADMIN OPTION clause.
- <User8> has been granted the SET USER system privilege to impersonate <User9> and <User10> with the <target_users_list> clause.
- <User9> as been granted the SET USER system privilege to impersonate <User10> with the <target_users_list> clause.

<User8> can successfully impersonate <User9> because:

- <User8> is granted the right to impersonate <User9> via the <target_users_list> clause (criteria 1).
- The <target_users_list> clause list for <User8> is a super-set of <User9> (criteria 4).
- Both <User8> and <User9> are granted <Role5>, with <User8> granted higher administrative rights to the role than <User9> (criteria 2 and 3).

<User8> can successfully impersonate <User10> because:

- <User8> is granted the right to impersonate <User10> (Criteria 1).
- Since <User10> is not granted the SET USER system privilege, requirement 4 is not applicable.
- Both <User8> and <User10> are granted <Role5>, with the same administrative rights to the role (criteria 2 and 3).

<User9> cannot impersonate <User8> because:

- <User9> is not granted the right to impersonate <User8> (Criteria 1.)

- Though both `<User8>` and `<User9>` are granted `<Role5>`, the grant for `<User9>` is with less administrative rights to the role than for `<User8>` (criteria 3).

Criterion are validated occurs when the SETUSER statement is executed, not when the SET USER system privilege is granted. If a user fails to meet any of the criteria when the SETUSER statement is issued, a `permission denied` message appears, and the impersonation does not begin.

1.5.2 Granting the SET USER System Privilege to a User

Allow one user to impersonate another user in the database. The system privilege can be granted with or without administrative rights.

Prerequisites

- The SET USER system privilege granted with administrative rights.
- Each target user specified (target_users_list) is an existing user or user-extended role with a login password.
- Each target role specified (target_roles_list) must be an existing user-extended or user-defined role.

Context

You can grant a user the ability to impersonate any user in the database (ANY), only specific users (`<target_users_list>`), or members of specific roles (ANY WITH ROLES `<target_roles_list>`). Administrative rights to the SET USER system privilege can be granted only when using the ANY clause.

If no clause is specified, ANY is the default.

When regranting the SET USER system privilege to a user, the effect of the grant is cumulative.

If no administrative clause is specified when using the ANY clause, WITH NO ADMIN OPTION is the default.

WITH NO ADMIN OPTION is the only valid administrative clause with the `<target_users_list>` or `<target_roles_list>` clauses.

Procedure

To grant the SET USER system privilege, execute one of these statements:

Grant Type	Statement
System privilege to impersonate any database user,	`GRANT SET USER(ANY)`

Grant Type	Statement
with full administrative rights	TO <user_ID [,...]> WITH ADMIN OPTION
System privilege to impersonate any database user, with administrative rights only	GRANT SET USER(ANY) TO <user_ID [,...]> WITH ADMIN ONLY OPTION
System privilege to impersonate any database user, with no administrative rights	GRANT SET USER(ANY) TO <user_ID [,...]> WITH NO ADMIN OPTION
System privilege to impersonate specified users	GRANT SET USER(<target_users_list>) TO <user_ID [,...]>
System privilege to impersonate any member of specified roles	GRANT SET USER(ANY WITH ROLES <target_roles_list>) TO <user_ID [,...]>
System privilege to impersonate specified users and members of specified roles	GRANT SET USER (<target_users_list>),(ANY WITH ROLES <target_roles_list>) TO <user_ID [,...]>

Example

Both of these statements grant <Sam> the ability to impersonate any database user:

```
GRANT SET USER (ANY) TO Sam
or
GRANT SET USER TO Sam
```

This statement grants <Bob> and <Jeff> the ability to impersonate <Mary>, <Joe>, or <Sue> only.

```
GRANT SET USER (Mary, Joe, Sue) TO Bob, Jeff
```

This statement grants <Mary> the ability to impersonate any member of the <Sales1> role:

```
GRANT SET USER (ANY WITH ROLES Sales1) TO Mary
```

This statement grants <Sarah> the ability to impersonate <Joe> or <Sue>, or any member of the <Sales2> role:

```
GRANT SET USER (Joe, Sue), (ANY WITH ROLES Sales2) TO Sarah
```

This statement grants <Joan> the ability to impersonate any member of the <Marketing1> or <Marketing2> roles:

```
GRANT SET USER (ANY WITH ROLES Marketing1, Marketing2) TO Joan
```

Related Information

GRANT SET USER Statement [page 327]

1.5.3 Starting to Impersonate Another User

Allows a user to assume the exact roles and system privileges (impersonate) of another user. Impersonation remains in effect until it is stopped or until the current session ends.

Prerequisites

The impersonator and target users meet all the requirements for impersonation. See *Understanding the Requirements for Impersonation*.

Context

At-least criteria is validated when the SETUSER command is executed, not when the SET USER system privilege is granted. When the SETUSER command is executed, if the impersonating user fails to meet all at-least criteria, a `permission denied` message appears, and impersonation does not begin. However, if all at-least criteria is met on a subsequent SETUSER execution, impersonation begins.

Once you issue the SETUSER statement, and impersonation begins, it remains in effect until you manually terminated the impersonation, begin impersonating another user, or the current session ends. While a user is impersonating another user, roles and privileges and their related administrative rights can be granted to or revoked from the impersonator or impersonatee as long as doing so does not violate the at-least criteria behind the impersonation. If the grant or revoke violates the criteria, an error message appears, and the statement fails. SAP recommends that impersonation be terminated as soon as the required tasks are complete.

Procedure

At a command prompt, type:

```
SETUSER <userID>
```

Related Information

Requirements for Impersonation [page 135]

SETUSER Statement [page 351]

1.5.4 Verifying the Current Impersonation Status of a User

A successful impersonation remains in effect until it is manually terminated or the session is terminated.

To verify the current status of an impersonation, execute this command on a machine on which the SETUSER command was issued:

```
SELECT CURRENT USER
```

This command returns the name of the user the machine recognizes as the currently logged in user. If it is the expected user for the machine, no impersonation is active on the machine. If an unexpected user name appears, it represents the user currently being impersonated on the machine.

Example

On a connection where Joe is logged in, execute:

```
> select current user
> go
current user
---------------------------------------------------------------------------------
-----------------------------------------------
Joe
(1 row affected)
>setuser mary
>go
>select current user
> go
current user
---------------------------------------------------------------------------------
-----------------------------------------------
Mary
```

1.5.5 Stopping Impersonation of Another User

End the impersonation of another user on the machine. Once begun, impersonation of another user remains in effect until impersonation is stopped, or the current session ends.

Prerequisites

The `SETUSER` command is issued from the same connection where it was initiated.

Procedure

At a command prompt, type:

```
SETUSER
```

Related Information

SETUSER Statement [page 351]

1.5.6 Revoking the SET USER System Privilege from a User

Remove the ability of a user to impersonate other users, and to administer the SET USER system privilege.

Prerequisites

The SET USER system privilege granted with administrative rights.

Context

The SET USER system privilege can be granted to a user multiple times, using different clauses. For example, `User1` is granted the SET USER system privilege once using the ANY clause and again with the target_users_list clause. In cases of multiple grants, the same form of the clause used for the GRANT must be used to revoke it. If the system privilege is revoked from `User1` using the ANY clause, the grant with the `<target_users_list>` clause remains in effect. The net effect is that User1 is now limited to impersonating users on the <target_users_list>. Alternately, if the system privilege is revoked from User1 using the <target_users_list> clause, the grant with the ANY clause remains in effect. The net effect in this scenario is that User1 can continue to impersonate any user in the database.

> i Note
>
> These examples assume User1 meets all criteria for successful impersonation.

Procedure

To revoke the SET USER system privilege, execute one of these statements:

Revoke Type	Description

Administrative rights to system privilege only	`REVOKE ADMIN OPTION FOR SET USER(ANY)`
	`FROM <user_ID [,...]>`

System privilege to impersonate any database user, including administrative rights	`REVOKE SET USER`
	FROM`FROM <user_ID [,...]>`

System privilege to impersonate specified users	`REVOKE SET USER(<target_users_list>)`
	`FROM <user_ID [,...]>`

System privilege to impersonate specified roles	`REVOKE SET USER(ANY WITH ROLES <target_roles_list>)`
	`FROM <user_ID [,...]>`

Example

These statements remove the ability for <Sam> to impersonate any database user:

```
REVOKE SET USER (ANY) FROM Sam
or
REVOKE SET USER FROM Sam
```

This statement removes administrative rights only to the SET USER system privilege from <Frank>. <Frank> can still impersonate any user in the database.

```
REVOKE ADMIN OPTION FOR SET USER (ANY) FROM Frank
```

This statement removes the ability of <Bob> and <Jeff> to impersonate <Mary>, <Joe>, or <Sue> only.

```
REVOKE SET USER (Mary, Joe, Sue) FROM Bob, Jeff
```

This statement removes the ability of `<Mary>` to impersonate any member of the `<Sales1>` role:

```
REVOKE SET USER (ANY WITH ROLES Sales1) FROM Mary
```

This statement removes the ability of `<Sarah>` to impersonate `<Joe>` or `<Sue>`, or any member of the `<Sales2>` role:

```
REVOKE SET USER (Joe, Sue), (ANY WITH ROLES Sales2) FROM Sarah
```

This statement removes the ability of `<Joan>` to impersonate any member of the `<Marketing1>` or `<Marketing2>` roles:

```
REVOKE SET USER (ANY WITH ROLES Marketing1, Markeing2) FROM Joan
```

Related Information

REVOKE SET USER Statement [page 342]

1.6 Users

User management includes the creation and deletion of user IDs, as well as password management.

In this section:

DBA User [page 146]
The DBA user is the default user created when a new SAP IQ database is created.

Super-User [page 148]
Super-users can exercise any system privilege and administer any role; they can perform any privileged operation in the system. Role-based security does not require a super-user to maintain the database; the DBA user might not be a super-user.

Increase Password Security [page 148]
Passwords are an important part of any database security system. There are several options for increasing password security.

Passwords in the Database [page 149]
As of version 15.0, SAP IQ uses SHA256 to hash passwords. Passwords are stored in UTF-8.

Case-Sensitivity of User IDs and Passwords [page 149]
Case-sensitivity of passwords is treated differently from other identifiers.

Creating a New User [page 150]
Create a new user ID.

Deleting a User [page 150]
Remove a user ID from the database.

Changing a User's Password [page 151]

Change the password of another user.

Converting a User-Extended Role Back to a User [page 152]
: You can convert a user-extended role back to a regular user.

Permanently Locking a User Account [page 153]
: To permanently lock a user account, you must assign a login policy with the locked option set to ON to the account. Once disabled, a user cannot connect to the SAP IQ server.

Unlocking User Accounts [page 154]
: Unlock a user account.

Automatic Unlocking of User Accounts [page 156]
: A lockdown of some or all database services may occur if all administrative users with the MANAGE ANY USER system privilege are locked out of the database due to failed login attempts.

1.6.1 DBA User

The DBA user is the default user created when a new SAP IQ database is created.

The password for the DBA user is initially set to "sql." To override the default user name or password during database creation, use the `CREATE DATABASE` statement with the DBA USER or DBA PASSWORD clause.

> i Note
>
> If you elect not to override the default password while creating the database, SAP strongly recommends that you do so as soon as possible thereafter.

By default, the DBA user is automatically granted administrative rights on the SYS_AUTH_DBA_ROLE role, which in turn is granted the SYS_AUTH_SA_ROLE and SYS_AUTH_SSO_ROLE roles. It is the union of these roles which grants the DBA user all system and object-level privileges in the database, and allows DBA to carry out any activity in the database: create tables, change table structures, create new user IDs, revoke privileges from users, and so on.

To ensure database security and accountability, avoid using generic names like "dba" as the first user ID. Use a real user's login name with a strong password instead.

Users Granted the SYS_AUTH_DBA_ROLE Role

Under certain circumstances, the underlying roles of SYS_AUTH_DBA_ROLE role can be dropped, and the underlying system privileges of the SYS_AUTH_SA_ROLE and SYS_AUTH_SSO_ROLE roles revoked. However, the SAP IQ documentation assumes that the DBA user is the database administrator, and all underlying roles and system privileges remain as granted by default.

To guard against password loss by the active DBA user, create one or more extra DBA accounts (with a randomly generated user name and password) and lock up those credentials. If the active DBA password is lost, use one of the extra credentials to log in to that DBA account, and reset the original account password.

Adding New Users

The DBA can add new users to the database. New users are then granted privileges to carry out authorized tasks on the database. Although DBA responsibilities may be handed over to other user IDs, the DBA is responsible for overall database management by virtue of the SYS_AUTH_DBA_ROLE role.

The DBA can then create database objects and assign ownership of these objects to other user IDs.

DBA User ID in Case-Sensitive Databases

User IDs and passwords are database objects.

In this section:

1.6.1.1 Changing the DBA Password

The default password for DBA user for all databases is `sql`. Change this password to prevent unauthorized access to your database.

Prerequisites

The CHANGE PASSWORD system privilege.

> → Tip
>
> If you are using `dbisql`, place your privilege grants into a command file for reference so you can modify and re-run it if necessary, to re-create the privileges.

Procedure

To change a user password, execute:

```
ALTER USER <userID>
IDENTIFIED BY <password>
```

Related Information

Case-Sensitivity of User IDs and Passwords [page 149]
ALTER USER Statement [page 294]

1.6.2 Super-User

Super-users can exercise any system privilege and administer any role; they can perform any privileged operation in the system. Role-based security does not require a super-user to maintain the database; the DBA user might not be a super-user.

By default, the DBA user can exercise any system privilege, but since it might not be able to administer all user-defined roles, it is not considered a true super-user. SAP IQ does not automatically create a super-user for a new or migrated database.

To create a super-user, create a user and grant it the SYS_AUTH_DBA_ROLE compatibility role.

> i Note
>
> If you migrated SYS_AUTH_DBA_ROLE, you must manually grant all of the underlying default system privileges of SYS_AUTH_DBA_ROLE, with administration rights, to create the super-user.

To maintain the super-user status, once you have created a super-user, all new user-extended and user-defined roles must be granted to the super-user, with administrative rights.

To allow the DBA user to act as a super-user, all new user-extended and user-defined roles must be granted to the DBA user, with administrative rights.

Administrative rights can be granted in the form of a role administrator or a global role administrator.

1.6.3 Increase Password Security

Passwords are an important part of any database security system. There are several options for increasing password security.

Implement a Login Policy control the frequency of password changes, to specify the number of login attempts allowed before an account is locked, or to force password expiration. See *Login Policies*.
Implement a Minimum Password Length by default, passwords can be any length. For greater security, you can enforce a minimum length requirement on all new passwords to disallow short (and therefore easily guessed) passwords. The recommended minimum length is 6. See *MIN_PASSWORD_LENGTH*.
Implement Password Rules implement advanced password rules that include requiring certain types of characters in the password, disallowing password reuse, and expiring passwords. Validation of the rules occurs when a new user ID is created or a password is changed. See *VERIFY_PASSWORD_FUNCTION*.

Related Information

Login Policies [page 156]
VERIFY_PASSWORD_FUNCTION Option [page 359]
MIN_PASSWORD_LENGTH Option [page 362]

1.6.4 Passwords in the Database

As of version 15.0, SAP IQ uses SHA256 to hash passwords. Passwords are stored in UTF-8.

When passwords are created or changed, they are converted to UTF-8 before being hashed and stored in the database. If the database is unloaded and reloaded into a database with a different character set, existing passwords continue to work. If the server cannot convert from the client's character set to UTF-8, SAP then recommends that passwords be composed of 7-bit ASCII characters as other characters may not work correctly.

1.6.5 Case-Sensitivity of User IDs and Passwords

Case-sensitivity of passwords is treated differently from other identifiers.

In SAP IQ and SAP SQL Anywhere, all passwords in newly created databases are case-sensitive, regardless of the case-sensitivity of the database. The default user ID is DBA and the password for this user is lowercase sql.

When you rebuild an existing database, SAP IQ and SAP SQL Anywhere determine the case-sensitivity of the password as follows:

- If the database was originally entered in a case-insensitive database, the password remains case-insensitive.
- If the password was originally entered in a case-sensitive database, uppercase and mixed-case passwords remain case-sensitive. If the password was entered in all lowercase, then the password becomes case-insensitive.
- Changes to both existing passwords and new passwords are case-sensitive.

In SAP Adaptive Server Enterprise (SAP ASE), the case-sensitivity of user IDs and passwords follows the case-sensitivity of the server.

1.6.6 Creating a New User

Create a new user ID.

Prerequisites

The MANAGE ANY USER system privilege.

Procedure

To create a new user, execute:

```
CREATE USER <userID>
IDENTIFIED BY <password>
```

Example

This statement adds user ID `Joe` to a database with password `welcome`:

```
CREATE USER Joe
IDENTIFIED BY welcome
```

Related Information

CREATE USER Statement [page 309]

1.6.7 Deleting a User

Remove a user ID from the database.

Prerequisites

- Requires the MANAGE ANY USER system privilege.
- The user being deleted does not own any database objects and is not currently connected to the database.

Context

If the user being delete has any external logins defined, the external logins are deleted as part of the process. However, any related objects on remote servers are not removed.

Procedure

To delete a user, execute:

```
DROP USER <userID>
```

> i Note
>
> - When dropping a user, any permissions granted by this user are also removed.
> - If the user being deleted owns any objects in the database, the following error message appears, and the command fails:
>
> ```
> Cannot drop a user that owns tables in runtime system SQLCODE=-128, ODBC 3
> State="42000"
>     Line 1, column 1
> ```

Example

This statement drops user ID `Joe` from the database:

```
DROP USER Joe
```

Related Information

DROP USER Statement [page 315]

1.6.8 Changing a User's Password

Change the password of another user.

Prerequisites

Requires the CHANGE PASSWORD system privilege.

Context

You can set up password rules (MIN_PASSWORD_LENGTH option) and verify that any new password assigned complies with them (VERIFY_PASSWORD_FUNCTION option). For example, you might require that passwords must include one digit or cannot be the user ID.

Procedure

To change a user password, execute:

```
ALTER USER <user_ID>
IDENTIFIED BY <password>
```

Example

This statement assigns the new password P&ssW0rd to user M_Smith:

```
ALTER USER M_Smith IDENTIFIED BY P&ssW0rd
```

Related Information

Case-Sensitivity of User IDs and Passwords [page 149]
Case-Sensitivity of User IDs and Passwords [page 149]
ALTER USER Statement [page 294]
VERIFY_PASSWORD_FUNCTION Option [page 359]
MIN_PASSWORD_LENGTH Option [page 362]

1.6.9 Converting a User-Extended Role Back to a User

You can convert a user-extended role back to a regular user.

Prerequisites

Administrative rights over the user-extended role being converted.

Context

The user retains any login privileges, system privileges, and roles that are granted to the user-extended role. The user remains as the owner of the objects that were created after the user was extended to act as a role. Any members of the user-extended role are immediately revoked.

A minimum number of role or global role administrators (as defined by the `MIN_ROLE_ADMINS` database option) with a login password must exist for each role at all times. When converting a user-extended role back to a user, all dependent roles of the user-extended role must continue to meet this minimum requirement, or the conversion fails.

Procedure

To convert a user-extended role back to a user, execute one of these:

Convert Condition	Statement
Role has not been granted any members.	`DROP ROLE FROM USER <role_name>`
Role has been granted members.	`DROP ROLE FROM USER <role_name> WITH REVOKE`

Related Information

DROP ROLE Statement [page 313]

1.6.10 Permanently Locking a User Account

To permanently lock a user account, you must assign a login policy with the locked option set to ON to the account. Once disabled, a user cannot connect to the SAP IQ server.

Prerequisites

- The MANAGE ANY LOGIN POLICY system privilege to create or alter the login policy.
- The MANAGE ANY USER system privilege to assign the login policy to users.

Procedure

1. Create a login policy with the LOCKED option set to ON.
2. Execute the `ALTER USER` command to assign the login policy to a user account to be disabled.

> i Note
>
> You cannot specify multiple user names in the same `ALTER USER` command when assigning a login policy to users.

Example

This command creates a new login policy named `lp_locked_users` with the `LOCKED` option set to ON:

```
CREATE LOGIN POLICY lp_locked_users locked=ON
```

These commands assign the `lp_locked_users` login policy to users `John` and `Mary`. `John` and `Mary` can no longer log in.

```
ALTER USER john LOGIN POLICY lp_locked_users
ALTER USER Mary LOGIN POLICY lp_locked_users
```

Related Information

ALTER USER Statement [page 294]
CREATE LOGIN POLICY Statement [page 300]

1.6.11 Unlocking User Accounts

Unlock a user account.

Prerequisites

Requires the MANAGE ANY USER system privilege.

Procedure

Do one of the following:

Reason for Account Lock	Task
User account is locked because it is assigned to a login policy with the locked option set to ON	Reassign the user to a login policy with the locked option set to OFF.
User account is locked because it has exceeded the MAX_FAILED_LOGIN_ATTEMPTS or MAX_DAYS_SINCE_LOGIN,	Issue the `ALTER USER` statement with the RESET LOGIN POLICY option. Forcing the reset of the login policy reverts the settings of the user's login to the original values in the login policy. This usually clears all locks that are implicitly set due to the user exceeding the failed number of logins, or exceeding the maximum number of days since the last login. i Note Resetting the values in the login policy assigned to a user does not reset the values for all users assigned the same login policy.

Example

Assuming that the LOCKED option in login policy `lp` is set to OFF, this example replaces the login policy currently assigned to `John` with login policy `lp`:

```
ALTER USER john LOGIN POLICY lp
```

Assuming `John's` account is locked because he either exceeded the MAX_FAILED_LOGIN_ATTEMPTS or MAX_DAYS_SINCE_LOGIN, this example forces the reset of the values in the login policy currently assigned to `John`:

```
ALTER USER john RESET LOGIN POLICY
```

Related Information

ALTER LOGIN POLICY Statement [page 284]
ALTER USER Statement [page 294]

1.6.12 Automatic Unlocking of User Accounts

A lockdown of some or all database services may occur if all administrative users with the MANAGE ANY USER system privilege are locked out of the database due to failed login attempts.

A user account is automatically locked if the user exceeds the maximum failed login attempts limit (MAX_FAILED_LOGIN_ATTEMPTS) value defined in the login policy. Once locked, the user account must be manually unlocked by a user who is granted the MANAGE ANY USER system privilege. However, if all users with the MANAGE ANY USER system privilege are locked out due to failed login attempts, a potential lockdown of some or all the database services can occur.

To prevent this scenario, use these login policy options:

> **ROOT_AUTO_LOCK_TIME** defines automatic unlocking period for users with the MANAGE ANY USER system privilege. You can set root_auto_lock_time to a small value (for example, 15 minutes) in the root login policy. There is a server-imposed upper limit of a few hours.
>
> **AUTO_UNLOCK_TIME** defines the automatic unlocking period for all other users. Set AUTO_UNLOCK_TIME to UNLIMITED (default) in any login policy, including the root login policy.

Configuration of these values requires the MANAGE ANY LOGIN POLICY system privilege.

Based on the permissions granted to a user, one of these login policy options is verified at the time of unlocking. Automatic unlocking is applicable only to locked accounts due to failed login attempts and not to accounts locked for any other reason. The locked status of a user is verified during login and if the user has equaled or exceeded the specified automatic unlock period, the user is allowed to log in and the FAILED_LOGIN_ATTEMPTS counter is reset to zero.

Related Information

Minimum Number of Role Administrators [page 31]
Minimum Number of Role Administrators [page 31]
ALTER LOGIN POLICY Statement [page 284]
ALTER USER Statement [page 294]
Permanently Locking a User Account [page 153]
Unlocking User Accounts [page 154]

1.7 Login Policies

A login policy defines the rules that SAP IQ follows to establish user connections. Each login policy is associated with a set of options called login policy options.

Login management commands that you execute on any multiplex server are automatically propagated to all servers in the multiplex. For best performance, execute these commands, or any DDL, on the coordinator.

In this section:

Modifying the Root Login Policy [page 157]
: You can modify the option values for the root login policy, but you cannot drop the policy.

Creating a New Login Policy [page 158]
: Any options that are not explicitly set when creating a login policy inherit their values from the root login policy.

Modifying an Existing Login Policy [page 159]
: Modify options within an existing login policy.

Deleting a Login Policy [page 160]
: You cannot delete the root login policy, or one that is currently assigned to a user.

Assigning a Login Policy When Creating a New User [page 160]
: If you do not assign a login policy when creating a user account, the account is assigned the root login policy.

Assigning a Login Policy to an Existing User [page 161]
: Assign a login policy to an existing SAP IQ user.

1.7.1 Modifying the Root Login Policy

You can modify the option values for the root login policy, but you cannot drop the policy.

Prerequisites

The MANAGE ANY LOGIN POLICY system privilege.

Context

Each new database is created with a default login policy, called the root policy. When you create a user account without specifying a login policy, the user becomes part of the root login policy.

Procedure

To modify the options of the root login policy, execute:

```
ALTER LOGIN POLICY ROOT <{login_policy_options}>
```

Related Information

ALTER LOGIN POLICY Statement [page 284]
Login Policy Options [page 302]
Multiplex Login Policy Configuration [page 290]
LDAP Login Policy Options [page 289]

1.7.2 Creating a New Login Policy

Any options that are not explicitly set when creating a login policy inherit their values from the root login policy.

Prerequisites

The MANAGE ANY LOGIN POLICY system privilege.

Context

Login policy names must be unique. You see an error message if the name of the login policy you are adding already exists.

Procedure

To create a new login policy, execute:

```
CREATE LOGIN POLICY <policy_name> <{login_policy_options}>
```

Example

This statement creates the `Test1` login policy with PASSWORD_LIVE_TIME option set to 60 days:

```
CREATE LOGIN POLICY Test1
password_life_time=60
```

Related Information

CREATE LOGIN POLICY Statement [page 300]
Login Policy Options [page 302]
Multiplex Login Policy Configuration [page 290]
LDAP Login Policy Options [page 289]

1.7.3 Modifying an Existing Login Policy

Modify options within an existing login policy.

Prerequisites

The MANAGE ANY LOGIN POLICY system privilege.

Procedure

To alter the options of an existing login policy, execute:

```
ALTER LOGIN POLICY <policy-name> <{login_policy_options}>
```

Example

This statement alters the LOCKED and MAX_CONNECTIONS options on the `Test1` login policy:

```
ALTER LOGIN POLICY Test1
locked=on
max_connections=5
```

Related Information

ALTER LOGIN POLICY Statement [page 284]
Login Policy Options [page 302]
Multiplex Login Policy Configuration [page 290]
LDAP Login Policy Options [page 289]

1.7.4 Deleting a Login Policy

You cannot delete the root login policy, or one that is currently assigned to a user.

Prerequisites

The MANAGE ANY LOGIN POLICY system privilege.

Procedure

1. Verify that no users are currently assigned the login policy to be dropped.
2. Execute:

   ```
   DROP LOGIN POLICY <policy_name>
   ```

Related Information

DROP LOGIN POLICY Statement [page 312]

1.7.5 Assigning a Login Policy When Creating a New User

If you do not assign a login policy when creating a user account, the account is assigned the root login policy.

Prerequisites

The MANAGE ANY LOGIN POLICY system privilege.

Context

Assign a login policy other than the root login policy when creating a new user. A user can be assigned only one login policy at a time.

Procedure

Execute:

```
CREATE USER <userID>
[ IDENTIFIED BY <password> ]
[ LOGIN POLICY <policy-name> ]
```

> Note
>
> You cannot specify multiple user IDs in the same `CREATE USER` command when assigning a login policy to users.

Example

This statement creates a user called `Joe` with the password `welcome`, and assigns the login policy `Test2`:

```
CREATE USER Joe
IDENTIFIED BY welcome
LOGIN POLICY Test2
```

Related Information

CREATE USER Statement [page 309]

1.7.6 Assigning a Login Policy to an Existing User

Assign a login policy to an existing SAP IQ user.

Prerequisites

The MANAGE ANY LOGIN POLICY system privilege.

Procedure

1. Execute:

   ```
   ALTER USER <userID>
   ```

```
LOGIN POLICY <policy_name>
```

2. Have the user log out and back in to apply the new login policy.

Related Information

ALTER USER Statement [page 294]

1.8 User Connections

There are several ways to manage user connections.

You can:

- Limit the number of active logins for a single user – assign user to a login policy in which the MAX_CONNECTIONS login policy option is set.
- Lock a user account:
  - Explicitly – assign user to a login policy in which the LOCKED option is set to ON.
  - Implicitly – assign user to a login policy in which the MAX_FAILED_LOGIN_ATTEMPTS option is set. If the user exceeds the value when attempting to log in, his or her user account is locked.
- Set a password expiry condition – assign user to a login policy in which the PASSWORD_EXPIRY_ON_NEXT_LOGIN login policy option is set. You can also execute the `CREATE USER` or `ALTER USER` statements, including the FORCE PASSWORD CHANGE clause.

Assigning a login policy to a user, or forcing a password change requires the MANAGE ANY USER system privilege. Creating or altering a login policy requires the MANAGE ANY LOGIN POLICY system privilege.

In this section:

1.8.1 Preventing Connection After Failed Login Attempts

Prevent a user from connecting after exceeding the maximum failed login attempts.

Prerequisites

- The MANAGE ANY LOGIN POLICY system privilege to create or alter the login policy.
- The MANAGE ANY USER system privilege to assign the login policy to users.

Context

You can set the system can be set to automatically lock an account if a user fails to enter valid login credentials after a specified number of attempts. Once locked, the user cannot connect, even if valid credentials are subsequently entered; the account remains locked until it is manually unlocked. The MAX_FAILED_LOGIN_ATTEMPTS login policy option controls the number of sequential failed attempts before the user account is locked. You can set this value in a new or existing login policy, including the root login policy, and it then applies to all users who are assigned the login policy.

Procedure

1. To set the MAX_FAILED_LOGIN_ATTEMPTS option, either create a new login policy, or modify an existing one.
2. Define a value for the MAX_FAILED_LOGIN_ATTEMPTS option.
3. Assign the login policy to applicable users, as needed.

Example

This example creates a new login policy named `lp`, which automatically locks a user account after 5 failed attempts:

```
CREATE LOGIN POLICY lp max_failed_login_attempts=5
```

This example modifies an existing login policy named `exist_lp`, which automatically locks a user account after 5 failed attempts:

```
ALTER LOGIN POLICY lp max_failed_login_attempts=5
```

This example assigns the login policy `lp` to user `John`. Once `John` is assigned the `lp` login policy, he cannot log in if he enters invalid credentials five times in sequence.

```
ALTER USER John LOGIN POLICY lp
```

Related Information

ALTER LOGIN POLICY Statement [page 284]
ALTER USER Statement [page 294]
CREATE LOGIN POLICY Statement [page 300]
Login Policy Options [page 286]
LDAP Login Policy Options [page 289]
Multiplex Login Policy Configuration [page 290]

1.8.2 Creating a DBA Recovery Account

Create a DBA recovery account for production systems. The DBA recovery account is a backup, in case you lose the original DBA account password.

Procedure

1. Create one or more extra DBA accounts, using randomly generated user names and passwords.
2. Lock the credentials in a secure location.

Related Information

CREATE USER Statement [page 309]

1.8.3 Logging In with a DBA Recovery Account

Log in using the DBA recovery account, and reset the original DBA account password.

Procedure

1. Retrieve the DBA recovery account user name and password from the secure location.
2. Log in using the recovery account.
3. Reset the original DBA account password.
4. Return the DBA recovery account credentials to their secure location.

1.8.4 Manage Connections Using Stored Procedures

There are several stored procedures for managing user connections.

This table lists the procedure available to perform each SAP IQ login management function.

Stored Procedure	Purpose	System Privilege Required
`sa_get_user_status`	Retrieve the current status of all existing users	MANAGE ANY USER system privilege to retrieve the current status of all existing users. Users without the MANAGE ANY USER system privilege can retrieve only their current status.
`sp_expireallpasswords`	Immediately expire all user passwords	MANAGE ANY USER system privilege
`sp_iqaddlogin`	Add users, define their passwords, specify login policy, and password expiry on next login	MANAGE ANY USER system privilege
`sp_iqcopyloginpolicy`	Create a new login policy by copying an existing one	MANAGE ANY LOGIN POLICY system privilege
`sp_iqdroplogin`	Drop the specified user	MANAGE ANY USER system privilege
`sp_iqmodifylogin`	Assign a given user to a login policy	MANAGE ANY USER system privilege
`sp_iqmodifyadmin`	Set an option on a named login policy to a certain value	MANAGE ANY LOGIN POLICY system privilege
`sp_iqpassword`	Change your own or another user's password	All users can run `sp_iqpassword` to change their own passwords. CHANGE PASSWORD system privilege is required to change the password of another user.

Related Information

sp_expireallpasswords System Procedure [page 381]

sp_iqcopyloginpolicy Procedure [page 392]
sp_iqdroplogin Procedure [page 402]
sp_iqmodifyadmin Procedure [page 411]
sp_iqmodifylogin Procedure [page 412]
sp_iqpassword Procedure [page 437]
sp_iqaddlogin Procedure [page 384]
sa_get_user_status system procedure [page 376]

1.8.5 Manage Resources Used by Connections

Building a set of users and roles allows you to manage permissions on a database. Another aspect of database security and management is to limit the resources an individual user can use.

For example, you may want to prevent a single connection from taking too much available memory or CPU resources, and slowing down other database users.

In this section:

1.8.5.1 Database Options That Govern User Resources

Database options that control resources are called resource governors. Set database options using the `SET OPTION` statement.

CURSOR_WINDOW_ROWS defines the number of cursor rows to buffer.

MAX_CARTESIAN_RESULT limits the number of result rows from a query containing a Cartesian join.

MAX_IQ_THREADS_PER_CONNECTION sets the number of processing threads available to a connection for use in IQ operations.

TEMP_CACHE_MEMORY_MB sets the size of the cache for the SAP IQ temporary store. (The server option `-iqtc` is the recommended way to set the temp cache size.)

QUERY_TEMP_SPACE_LIMIT limits the amount of temporary dbspace available to any one query.

QUERY_ROWS_RETURNED_LIMIT tells the query optimizer to reject queries that might consume too many resources. If the optimizer estimates that the result set from the query will exceed the value of this option, the optimizer rejects the query and returns an error message.

The following database options affect the engine, but have limited impact on SAP IQ:

JAVA_HEAP_SIZE sets the maximum size (in bytes) of the memory allocated to Java applications on a per connection basis.

MAX_CURSOR_COUNT limits the number of cursors for a connection.

MAX_STATEMENT_COUNT limits the number of prepared statements for a connection.

Database option settings are not inherited through the role structure.

Related Information

SET OPTION Statement [page 348]

1.9 Security with Views and Procedures

You can use views and stored procedures to tailor privileges to suit the needs of your enterprise.

For databases that require a high level of security, there are limitations on defining privileges directly on tables. Any privilege granted to a user on a table applies to the entire table. You may need to assign privileges more precisely than on a table-by-table basis. For example:

- You do not want to give access to personal or sensitive information stored in an employee table to users who need access to other parts of the table.
- You may want to give sales representatives privileges on a table containing descriptions of sales calls, but only allow them to update privileges to their own calls.

In this section:

Views Provide Tailored Security [page 167]
Use views to give users access to only one portion of a table.

Use Procedures to Provide Tailored Security [page 170]
Procedures restrict the actions a user may take.

1.9.1 Views Provide Tailored Security

Use views to give users access to only one portion of a table.

You can define a portion in terms of rows or columns. For example, you may want to disallow a group of users from seeing the `Salary` column of an `Employees` table, or you may want to allow a user to see only the rows of a table that he or she have created.

Example 1

The sales manager needs access to information in the database concerning employees in the department. However, there is no reason for the manager to have access to information about employees in other departments.

Create a user ID for the sales manager, create views that provide the information needed, and grant the appropriate privileges to the sales manager user ID.

1. As a user with the MANAGE ANY USER system privilege, create the new user ID using the `GRANT` statement. Enclose DBA in quotation marks, because it is a SQL keyword.

   ```
   CONNECT "DBA"
   IDENTIFIED by sql;
   GRANT CONNECT
   TO SalesManager
   IDENTIFIED BY sales
   ```

2. Define a view that looks only at sales employees. Identify the table as `“DBA”.Employees`, with the owner of the table explicitly identified, so that the `SalesManager` user ID can use the view. Otherwise, when `SalesManager` uses the view, the `SELECT` statement refers to a table that the user ID does not recognize.

   ```
   CREATE VIEW emp_sales AS
   SELECT EmployeeID, GivenName, Surname
   FROM "DBA".Employees
   WHERE DepartmentID = 200
   ```

3. Give SalesManager privilege to look at the view. Use the same command to grant privilege on a view as to grant privilege on a table.

   ```
   GRANT SELECT
   ON emp_sales
   TO SalesManager
   ```

Example 2

This example creates a view, which allows the sales manager to look at a summary of sales orders. This view requires information from more than one table for its definition:

1. Create the view.

   ```
   CREATE VIEW order_summary AS
   SELECT OrderDate, Region, SalesRepresentative
   FROM "GROUPO".SalesOrders
       KEY JOIN "GROUPO".Customers
   ```

2. Grant privilege for `SalesManager` to examine this view.

   ```
   GRANT SELECT
   ON order_summary
   TO SalesManager
   ```

3. To check that the process has worked properly, connect to the `SalesManager` user ID and look at the views you have created:

   ```
   CONNECT SalesManager IDENTIFIED BY sales ;
   SELECT * FROM "GROUPO".emp_sales ;
   SELECT * FROM "GROUPO".order_summary ;
   ```

No privileges have been granted to `SalesManager` to look at the underlying tables. Therefore, these commands produce privilege errors:

```
SELECT * FROM "DBA".Employees ;
```

```
SELECT * FROM "DBA".SalesOrders;
```

These examples show how to use views to tailor SELECT privileges. You can grant INSERT, DELETE, and UPDATE privileges on views in the same way.

In this section:

1.9.1.1 Guidelines for Using Views

There are certain restrictions, both on the SELECT statements you use to create views, and on your ability to insert into, delete from, or update them.

Restrictions on SELECT Statements

You cannot use an ORDER BY clause in the SELECT query. A characteristic of relational tables is that there is no significance to the ordering of the rows or columns, and using an ORDER BY clause imposes an order on the rows of the view. You can use the GROUP BY clause, subqueries, and joins in view definitions.

Scalar value subqueries are supported only within the top-level SELECT list (not in a view, a derived table, or a subquery). Sometimes views or derived tables used in the FROM clause of the top-level SELECT are simple enough that they can be "flattened" up into the top-level SELECT. As a result of this, the preceding rule is actually enforced only for subqueries, nonflattened views, and nonflattened derived tables. For example:

```
CREATE VIEW test_view AS SELECT testkey,(SELECT COUNT(*) FROM tagtests WHERE
tagtests.testkey = testtrd.testkey ) FROM testtrd
```

```
SELECT * FROM test_view
Msg 21, Level 14, State 0:
SQL Anywhere Error -1005004: Subqueries are allowed only as arguments of
comparisons, IN, and EXISTS,
-- (opt_Select.cxx 2101)
```

To develop a view, tune the SELECT query by itself until it provides exactly the results you need in the format you want. Once you have the correct SELECT query, you can add a phrase in front of the query to create the view. For example:

```
CREATE VIEW <viewname> AS
```

Guidelines for Inserting and Deleting from Views

UPDATE, INSERT, and DELETE statements are allowed on some views, but not on others, depending on their associated SELECT statement.

You cannot update, insert into, or delete from views that contain:

- Aggregate functions, such as `COUNT(*)`
- A GROUP BY clause in the `SELECT` statement
- A `UNION` operation

In all these cases, there is no way to translate the `UPDATE`, `INSERT`, or `DELETE` into an action on the underlying tables.

> ⚠ Caution
>
> Do not delete views owned by the dbo user ID, which owns system objects. Deleting such views or changing them into tables may cause unexpected problems.

1.9.2 Use Procedures to Provide Tailored Security

Procedures restrict the actions a user may take.

A user may have EXECUTE privilege on a procedure without having any privileges on the table or tables on which the procedure acts.

By default, procedures execute with the privileges of the procedure owner. For a procedure that updates a table, if the procedure owner has UPDATE privileges on the table, the user can execute the procedure. The owner of the procedure can restrict the procedure to execute with the privileges of the user executing the procedure by specifying SQL SECURITY INVOKER to a CREATE/ALTER PROCEDURE statement.

In this section:

Setting Up Task-Based Security Restrictions [page 171]
Disallow all access to the underlying tables, and grant privileges to users or roles to execute certain stored procedures. This approach strictly defines how to control database modifications.

Granting Users the Privilege to Run Related Stored Procedures [page 171]
Grant users the system privilege required to run stored procedures. Since most privileges are inherited through role membership, users can inherit the system privilege and the execute privileges for IQ procedures from a role.

1.9.2.1 Setting Up Task-Based Security Restrictions

Disallow all access to the underlying tables, and grant privileges to users or roles to execute certain stored procedures. This approach strictly defines how to control database modifications.

Context

To allow users with specific privileges to administer certain tasks using SAP IQ system procedures:

Procedure

1. Create a role for each set of authorized tasks to be performed, and grant the role the applicable system privileges.
2. Grant each of these roles to a single common role.
3. Grant EXECUTE privileges on the IQ procedure for performing the authorized tasks to the applicable role.
4. When you create a new user who is to be granted authorized tasks, grant the role created for each authorized task to the user.

1.9.2.2 Granting Users the Privilege to Run Related Stored Procedures

Grant users the system privilege required to run stored procedures. Since most privileges are inherited through role membership, users can inherit the system privilege and the execute privileges for IQ procedures from a role.

Prerequisites

The MANAGE ANY USER or EXECUTE ANY PROCEDURE system privilege.

Context

To grant user `user1` the MANAGE ANY USER system privilege and privileges to execute procedures related to user administration:

Procedure

1. Create a role `USER_ADMIN_GRP`:

   ```
   CREATE ROLE USER_ADMIN_GRP
   ```

2. Grant the MANAGE ANY USER system privilege to the `USER_ADMIN_GRP` role:

   ```
   GRANT MANAGE ANY USER TO USER_ADMIN_GRP
   ```

3. Grant EXECUTE privilege on SAP IQ stored procedures for user administration to `USER_ADMIN_GRP`:

   ```
   GRANT EXECUTE on sp_iqaddlogin
   to USER_ADMIN_GRP
   GRANT EXECUTE on sp_iqcopyloginpolicy
   to USER_ADMIN_GRP
   GRANT EXECUTE on sp_iqdroplogin
   to USER_ADMIN_GRP
   GRANT EXECUTE on sp_iqmodifyadmin
   to USER_ADMIN_GRP
   GRANT EXECUTE on sp_iqmodifylogin
   to USER_ADMIN_GRP
   ```

4. Grant the `USER_ADMIN_GRP` role to `user1`. `user1` inherits the MANAGE ANY USER system privilege and the ability to execute the assigned IQ procedures through membership in `USER_ADMIN_GRP` role.

   ```
   GRANT ROLE USER_ADMIN_GRP TO user1
   ```

In this section:

Related Stored Procedures for Role Access [page 173]
You may create roles that grant privileges for various related stored procedures.

1.9.2.2.1 Related Stored Procedures for Role Access

You may create roles that grant privileges for various related stored procedures.

Role Name	System Privilege Granted	Stored Procedure
OPERATOR_GRP	BACKUP DATABASE DROP CONNECTION CHECKPOINT MONITOR ACCESS SERVER LS	sp_iqbackupdetails sp_iqbackupsummary sp_iqconnection sp_iqsysmon
SPACEADMIN_GRP	MANAGE ANY DBSPACE ACCESS SERVER LS	sp_iqdbspace sp_iqdbspaceinfo sp_iqdbspaceobjectinfo sp_iqemptyfile sp_iqestdbspaces sp_iqfile sp_iqobjectinfo sp_iqspaceused

Related Information

sp_iqbackupdetails Procedure [page 386]
sp_iqbackupsummary Procedure [page 388]
sp_iqconnection Procedure [page 389]
sp_iqdbspace Procedure [page 393]
sp_iqdbspaceinfo Procedure [page 396]
sp_iqdbspaceobjectinfo Procedure [page 399]
sp_iqemptyfile Procedure [page 403]
sp_iqestdbspaces Procedure [page 406]
sp_iqfile Procedure [page 407]
sp_iqobjectinfo Procedure [page 413]
sp_iqspaceused Procedure [page 416]
sp_iqsysmon Procedure [page 418]

1.10 Data Confidentiality

You can secure communications between a client and the SAP IQ server, or between an SAP IQ client and the database server using Transport Layer Security (TLS).

SAP IQ allows you to encrypt your database or columns.

Support of Kerberos authentication, and column encryption is included in the separately licensed SAP IQ Advanced Security Option.

In this section:

Database encryption and decryption [page 174]
  You can encrypt your database to make it more difficult for someone to decipher the data in your database.

IPv6 Support [page 180]
  SAP IQ supports Internet Protocol version 6 (IPv6), which contains addressing and control information to route packets over the Internet.

How to Set Up Transport Layer Security [page 180]
  Set up transport layer security on your system.

Digital certificates [page 181]
  You need digital certificates to set up transport layer security.

Related Information

Column Encryption in SAP IQ [page 241]

1.10.1 Database encryption and decryption

You can encrypt your database to make it more difficult for someone to decipher the data in your database.

You can choose to secure your database either with simple or with strong encryption.

> i Note
>
> If your database is encrypted, compressing it with a tool such as WinZip does not result in a file that is significantly smaller than the original database file.

In this section:

Simple encryption and strong encryption [page 175]
  Simple encryption and strong encryption imply the degree of accessibility for database-related information.

How to encrypt a database [page 176]

There are several ways to encrypt a database.

Creating an encrypted database (SQL) [page 176]
Encrypt a database during creation.

Creating an encrypted database (iqinit utility) [page 178]
Encrypt a database during creation.

Encryption keys [page 179]
Encryption keys can be of arbitrary length. Longer keys are better because a shorter key is easier to guess.

Performance issues when using encryption [page 179]
Performance is slower when the database is encrypted. The performance impact depends on how often pages are read from or written to disk, and can be minimized by ensuring that the server is using an adequate cache size.

1.10.1.1 Simple encryption and strong encryption

Simple encryption and strong encryption imply the degree of accessibility for database-related information.

Simple encryption

Simple encryption is equivalent to obfuscation and makes it more difficult for someone using a disk utility to look at the file to decipher the data in your database. Simple encryption does not require a key to encrypt the database.

Strong encryption

Strong database encryption technology makes a database inoperable and inaccessible without a key (password). An algorithm encodes the information contained in your database and transaction log files so they cannot be deciphered.

The database administrator has control over four aspects of strong encryption, including:

- strong encryption status
- encryption key
- protection of the encryption key
- encryption algorithm

Supported strong encryption algorithms

The algorithm used to implement strong encryption is AES: a block encryption algorithm chosen as the new Advanced Encryption Standard (AES) for block ciphers by the National Institute of Standards and Technology (NIST).

You can also specify a separate FIPS-approved AES module for strong encryption using the AES_FIPS (128-bit) or AES256_FIPS (256-bit) type. When the database server is started with the -fips option, you can run databases encrypted with AES, AES256, AES_FIPS, or AES256_FIPS strong encryption, but not databases encrypted with simple encryption. Unencrypted databases can also be started on the server when -fips is specified.

The SAP IQ security option must be installed on any computer used to run a database encrypted with AES_FIPS or AES256_FIPS.

FIPS-certified encryption requires a separate license. All strong encryption technologies are subject to export regulations.

1.10.1.2 How to encrypt a database

There are several ways to encrypt a database.

To create an encrypted database

You can use the following:

- The Initialization utility (iqint), with options such as -ep, -ek, or -ea to enable strong encryption.
- CREATE DATABASE statement.

You cannot encrypt an existing database.

1.10.1.3 Creating an encrypted database (SQL)

Encrypt a database during creation.

Prerequisites

By default, you must have the SERVER OPERATOR system privilege. The required privileges can be changed by using the -gu database server option.

Context

> ⚠ Caution
>
> For strongly encrypted databases, store a copy of the key in a safe location. If you lose the encryption key, there is no way to access the data, even with the assistance of Technical Support. The database must be discarded and you must create a new database.

Procedure

1. In Interactive SQL, connect to an existing database.
2. Execute a CREATE DATABASE statement that includes the ENCRYPTED clause and the KEY and ALGORITHM options.

Results

An encrypted database is created.

Example

For example, the following statement creates a database file named `myencrypteddb.db` in the `c:\temp\` directory using FIPS-certified 128-bit AES encryption and the specified encryption key. The DBA user ID and password are specified and a transaction log is enabled.

```
CREATE DATABASE 'c:\\temp\\myencrypteddb.db'
DBA USER 'DBA'
DBA PASSWORD 'passwd'
TRANSACTION LOG ON
ENCRYPTED ON
  KEY '0kZ2o52AK#'
  ALGORITHM 'AES_FIPS';
```

1.10.1.4 Creating an encrypted database (iqinit utility)

Encrypt a database during creation.

Prerequisites

There are no prerequisites for this task.

Context

> ⚠ Caution
>
> For strongly encrypted databases, store a copy of the key in a safe location. If you lose the encryption key, there is no way to access the data, even with the assistance of Technical Support. The database must be discarded and you must create a new database.

Procedure

Run the iqinit utility to create a database.

- ○ To encrypt the database with simple encryption, include the -ea simple option.
- ○ To encrypt the database with strong encryption, include -ek or -ep options to specify the encryption key.

Results

An encrypted database is created.

Example

- The following example creates the database `test.db` using simple encryption, and configures the DBA user ID as DBA with password sql:

  ```
  iqinit -dba DBA,sql -ea simple test.db
  ```

- The following command creates a strongly encrypted database and specifies the encryption key and algorithm.

  ```
  iqinit -dba DBA,sql -ek "0kZ2o56AK#" -ea AES_FIPS "myencrypteddb.db"
  ```

To start this database, run the following command:

```
start_iq myencrypteddb.db -ek "0kZ2o56AK#"
```

Next Steps

When starting or connecting to the database, you must specify the encryption key.

1.10.1.5 Encryption keys

Encryption keys can be of arbitrary length. Longer keys are better because a shorter key is easier to guess.

It is best to choose an encryption key value that cannot be easily guessed. As well, including a combination of numbers, letters, and special characters decreases the chances of someone guessing the key.

Encryption keys are always case sensitive, and they cannot contain leading or trailing spaces or semicolons.

You must supply this key each time you want to start the database. Lost or forgotten keys result in completely inaccessible databases.

You can choose whether the encryption key is entered at a command prompt (the default) or into a prompt box. Choosing to enter the key in a prompt box provides an extra measure of security because the key is never visible in plain sight. Clients are required to specify the key each time they start the database. If the database administrator starts the database, clients never need to have access to the key.

> ⚠ Caution
>
> For strongly encrypted databases, store a copy of the key in a safe location. If you lose the encryption key, there is no way to access the data, even with the assistance of Technical Support. The database must be discarded and you must create a new database.

1.10.1.6 Performance issues when using encryption

Performance is slower when the database is encrypted. The performance impact depends on how often pages are read from or written to disk, and can be minimized by ensuring that the server is using an adequate cache size.

You can increase the starting size of the cache with the -c option when you start the server. For operating systems that support dynamic resizing of the cache, the cache size that is used may be restricted by the amount of memory that is available; to increase the cache size, increase the available memory.

1.10.2 IPv6 Support

SAP IQ supports Internet Protocol version 6 (IPv6), which contains addressing and control information to route packets over the Internet.

IPv6 supports two^{128} unique IP addresses, which is a substantial increase over the number of addresses supported by its predecessor IPv4. SAP IQ supports both IPv4 and IPv6 addresses anywhere you can specify an IP address on the client or server.

JDBC and ODBC classes support the use of IPv6 addresses for remote data access.

1.10.3 How to Set Up Transport Layer Security

Set up transport layer security on your system.

Prerequisites

Obtain digital certificates. You need identity files and certificate files. The server identity file contains the server's private key and should be stored securely with the database. You distribute the server certificate file to your clients.

You can buy certificates from a certificate authority or you can use the Certificate creation utility (createcert). Functionality to create certificates, which is especially useful for development and testing, is provided.

Procedure

1. If you are setting up transport layer security for SAP IQ client/server applications:

   Start the SAP IQ database server with transport layer security

   Use the -ec database server option to specify the type of security, the server identity file name, and the password to protect the server's private key.

   If you also want to allow unencrypted connections over shared memory, specify the -es option.

   TDS connections do not use the TLS protocol. To prevent unencrypted connections from using the TDS protocol, specify the tcpip option -x tcpip(TDS=NO).

   Configure client applications to use transport layer security

   Specify the path and file name of trusted certificates using the Encryption connection parameter [ENC].
2. If you are setting up transport layer security for SAP IQ web services:

   Start the SAP IQ database server with transport layer security

   Use the -xs database server option to specify the type of security, the server identity file name, and the password to protect the server's private key.

Configure browsers or other web clients to trust certificates

Encrypt SAP IQ web services.

3. If you are setting up an SAP IQ multiplex database server:
    - INC, MIPC, and DAS connections determine which TLS connection parameters to use from the contents of the -ec server option.
    - Set the TRUSTED_CERTIFICATES_FILE option to the appropriate Certificate Authority.

Results

You have set up transport layer security on your system.

1.10.4 Digital certificates

You need digital certificates to set up transport layer security.

Obtain certificates from your computer's operating system certificate store or a certificate authority, or create them using the Certificate Creation utility (createcert).

Certificates from the operating system certificate store

By default, secure connections use your computer's operating system certificate store to obtain a trusted certificate for secure connections.

For all secure connections except HTTPS on Windows operating systems, stored certificates are cached, with the cache reloading every 24 hours. If a required certificate is installed within 24 hours after the first secure connection, then the connection requiring that certificate fails until the cache is reloaded. To make the certificate accessible before 24 hours, restart the server.

Certificate Creation utility

Use the Certificate Creation utility (createcert) to generate X.509 certificate files using RSA.

Certificate Viewer utility

Use the Certificate Viewer utility (viewcert) to read X.509 certificates using RSA.

Certificates for server authentication

Follow the same process to create certificate files for server authentication. In each case, create an identity file and a certificate file.

For server authentication, you can create a server identity file and a certificate file to distribute to clients.

Certificate configurations

The certificate can be self-signed or signed by a commercial or enterprise Certificate Authority.

Self-signed certificates

Self-signed server certificates can be used for simple setups.

Enterprise root certificates

An enterprise root certificate can be used to sign server certificates to improve data integrity and extensibility for multi-server deployments.

- You can store the private key used to sign server certificates in a secure central location.
- For server authentication, you can add database servers without reconfiguring clients.

Commercial Certificate Authorities

You can use a third-party Certificate Authority instead of an enterprise root certificate. Commercial Certificate Authorities have dedicated facilities to store private keys and create high-quality server certificates.

In this section:

Self-signed root certificates [page 182]

Self-signed root certificates can be used for simple setups involving a single MobiLink or database server.

Certificate chains [page 183]

If you require multiple identity files, you can improve security and extensibility by using certificate chains instead of self-signed certificates.

Globally signed certificates [page 185]

Globally signed certificates are high-quality certificates that are used to sign your certificate requests.

1.10.4.1 Self-signed root certificates

Self-signed root certificates can be used for simple setups involving a single MobiLink or database server.

Self-signed root certificates can be used for simple setups involving a single database server.

→ Tip

Use enterprise level certificate chains or commercial certificate authorities if you require multiple server identity files. Certificate authorities provide extensibility and a higher level of certificate integrity with dedicated facilities to store root private keys.

Certificate

For server authentication certificates, the self-signed certificate is distributed to clients. It is an electronic document including identity information, the public key of the server, and a self-signed digital signature.

Identity file

For server authentication certificates, the identity file is stored securely with a database server. It is a combination of the self-signed certificate (that is distributed to clients) and the corresponding private key. The private key gives the database server the ability to decrypt messages sent by the client in the initial handshake.

1.10.4.2 Certificate chains

If you require multiple identity files, you can improve security and extensibility by using certificate chains instead of self-signed certificates.

Certificate chains require a Certificate Authority or an enterprise root certificate to sign identities.

Benefits of using certificate chains

Certificate chains provide the following advantages:

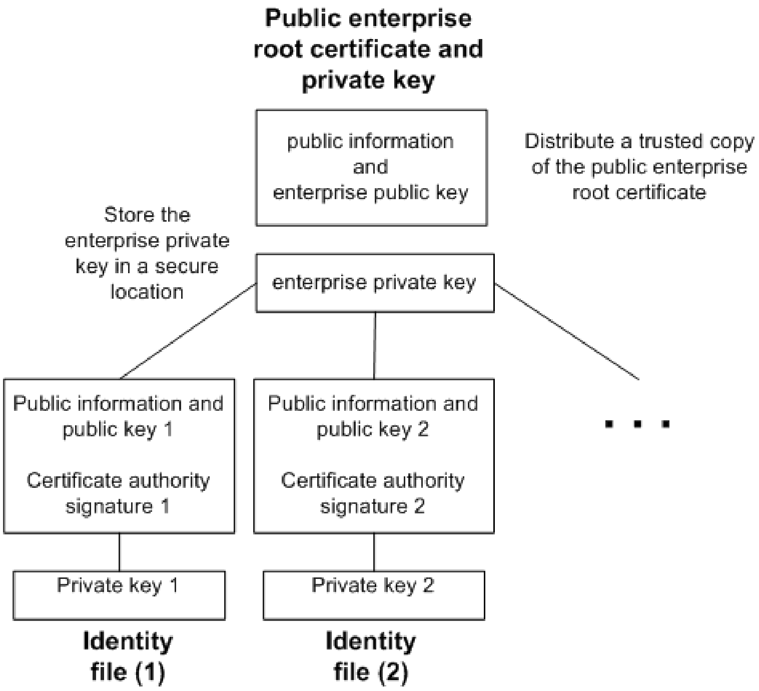
Extensibility

For server authentication, you can configure clients to trust any certificate signed by an enterprise root certificate or Certificate Authority. If you add a new database server, clients do not require a copy of the new certificate.

Security

The enterprise root certificate's private key is not in the identity file. Storing the root certificate's private key in a high-security location, or using a Certificate Authority with dedicated facilities, protects the integrity of server authentication.

The following diagram provides the basic enterprise root certificate architecture.


Public enterprise root certificate and private key
public information and enterprise public key
Distribute a trusted copy of the public enterprise root certificate
Store the enterprise private key in a secure location
enterprise private key
Public information and public key 1
Certificate authority signature 1
Public information and public key 2
Certificate authority signature 2
. . .
Private key 1
Private key 2
Identity file (1)
Identity file (2)

Using certificates in a multi-server environment

To create certificates used in a multi-server environment:

- Generate a public enterprise root certificate and enterprise private key.
  Store the enterprise private key in a secure location, preferably a dedicated facility.
  For server authentication, you distribute the public enterprise root certificate to clients.
- Use the enterprise root certificate to sign identities.
  Use the public enterprise root certificate and enterprise private key to sign each identity. For server authentication, the identity file is used for the server.

You can also use a third-party Certificate Authority to sign your server certificates. Commercial Certificate Authorities have dedicated facilities to store private keys and create high-quality server certificates.

In this section:

1.10.4.2.1 Enterprise root certificates

Enterprise root certificates improve data integrity and extensibility for multi-server deployments.

- You can store the private key used to create trusted certificates in a dedicated facility.
- For server authentication, you can add servers without reconfiguring clients.

To set up enterprise root certificates, you create the enterprise root certificate and the enterprise private key that you use to sign identities.

1.10.4.2.2 Signed identity files

You can use an enterprise root certificate to sign server identity files.

For server authentication, you generate identity files for each server. Since these certificates are signed by an enterprise root certificate, you use the createcert -s option.

1.10.4.3 Globally signed certificates

Globally signed certificates are high-quality certificates that are used to sign your certificate requests.

These high-quality certificates are created and managed by a commercial Certificate Authority.

Globally signed certificates have the following advantages:

- For inter-company communication, common trust in an outside, recognized authority may increase confidence in the security of the system. A Certificate Authority must guarantee the accuracy of the identification information in any certificate that it signs.
- Certificate Authorities provide controlled environments and advanced methods to generate certificates.
- The private key for the root certificate must remain private. Your organization may not have a suitable place to store this crucial information, whereas a Certificate Authority can afford to design and maintain dedicated facilities.

Setting up globally signed certificates

To set up globally signed identity files, you:

- Create a certificate request using the createcert utility with the -r option.
- Use a Certificate Authority to sign each request. You can combine the signed request with the corresponding private key to create the server identity file.

> i Note
>
> You might be able to globally sign an enterprise root certificate. This is only applicable if your Certificate Authority generates certificates that can be used to sign other certificates.

In this section:

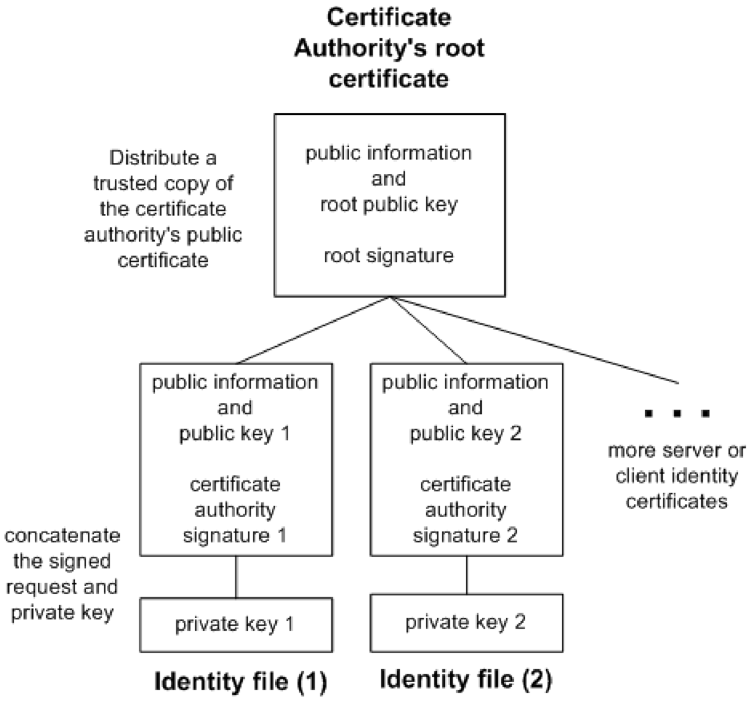
Globally signed identity files [page 186]
: You can use globally signed certificates directly as server identity files.

Client trust setup for the certificate authority's certificate [page 186]
: For server authentication, you must ensure that clients contacting your server trust the root certificate in the chain.

1.10.4.3.1 Globally signed identity files

You can use globally signed certificates directly as server identity files.

The following diagram shows the configuration for multiple identity files:


Certificate Authority's root certificate
Distribute a trusted copy of the certificate authority's public certificate
public information and root public key
root signature
public information and public key 1
certificate authority signature 1
public information and public key 2
certificate authority signature 2
. . .
more server or client identity certificates
concatenate the signed request and private key
private key 1
private key 2
Identity file (1)
Identity file (2)

You reference the server identity file and the password for the private key on the start_iq command line.

1.10.4.3.2 Client trust setup for the certificate authority's certificate

For server authentication, you must ensure that clients contacting your server trust the root certificate in the chain.

For globally signed certificates, the root certificate is the Certificate Authority's certificate.

When using a globally signed certificate, each client must verify certificate field values to avoid trusting certificates that the same Certificate Authority has signed for other clients.

1.11 Utility Database Server Security

SAP IQ includes a phantom database, called the utility database, that has no physical representation, and which can contain no data.

The utility database can run on any SAP IQ server. In SAP IQ Cockpit, the server for the utility database is known as the Utility Server.

The utility database permits a narrow range of specialized functions. It enables you to execute database file manipulation statements such as `CREATE DATABASE` and `DROP DATABASE` without first connecting to a physical database.

You can also retrieve database and connection properties from the utility database. These properties apply to databases you create when connected to the utility database.

One of your configuration tasks is to set up security for the utility database and its server. You must decide:

- Who can connect to the utility database, and
- Who can execute file administration statements.

In this section:

1.11.1 Defining the Utility Database Name When Connecting

You cannot specify a database file when starting the utility database, because no database file is associated with that database. You must specify the database name when connecting.

Procedure

Specify `utility_db` as the database name when connecting to the utility database.

For example:

```
dbisqlc -c "uid=dba;pwd=sql;eng=myserver;dbn=utility_db"
```

Results

> **Note**
>
> When you connect to the utility database to create an IQ database that uses Windows raw partitions, there is a syntax difference in the IQ PATH. For example, to specify a Windows raw partition on device I: for the utility database, you can use the specification “\\.\I:” On other IQ databases, you must double the slash characters, so that the same device is specified as “\\\\.\\I:”. The backslash character is treated as an escape character in IQ databases but as a normal character in the utility database.

1.11.2 Defining the Utility Database Password

Define the user ID `DBA` for the utility database.

Procedure

1. Use a text editor to open the file `util_db.ini`, which is stored in the server executable directory.

   Because this directory is on the server, you can control access to the file, and thereby also control who has access to the password.
2. Locate this line, and replace "password" with the password you want to use:

   ```
   [UTILITY_DB]
   PWD=password
   ```

   Use of the utility_db security level relies on the physical security of the computer hosting the database server, since the `util_db.ini` file can be easily read using a text editor.

1.11.3 Permission to Execute File Administration Statements

A separate level of security, which controls the creating and dropping of databases, provides additional database security. The `-gu` database server command line option controls who can execute the file administration statements.

There are four levels of permission for the use of file administration statements: `all`, `none`, `DBA`, and `utility_db`. The `utility_db` level permits a user who can connect to the utility database to use the file administration statements.

Table 1: Permissions for Role Administration

-gu Switch Value	Effect	Applies To
all	Anyone can execute file administration statements	Any database including the utility database
none	No one can execute file administration statements	Any database including the utility database
DBA	Only users with the SERVER OPERATOR system privilege can execute file administration statements	Any database including the utility database
utility_db	Only the users who can connect to the utility database can execute file administration statements	Only the utility database

Examples

On Sun, HP, Linux, and Windows platforms, to permit only the user knowing the utility database password to connect to the utility database and create or delete databases, start the server at the command line:

```
start_iq -n testsrv -gu utility_db
```

On AIX, to permit only the user knowing the utility database password to connect to the utility database and create or delete databases, start the server at the command line:

```
start_iq -n testsrv -gu utility_db -iqmt 256
```

Assuming that the utility database password was set to `IQ&Mine49` during installation, this command starts the Interactive SQL utility as a client application, connects to the server named `testsrv`, loads the utility database, and connects the user:

```
dbisql -c "uid=DBA;pwd=IQ&Mine49;dbn=utility_db;eng=testsrv"
```

Executing this statement successfully connects you to the utility database, and you can now create and delete databases.

> i Note
>
> The database name, user ID, and password are case-sensitive. Make sure that you specify the same case in the `dbisql` command and the `util_db.ini` file.

1.12 Data Security

Since databases may contain proprietary, confidential, or private information, it is important that you ensure that the database and the data in it are designed for security.

In this section:

1.12.1 System Secure Features

You can make system secure features inaccessible to databases running on a database server.

When a feature is secured (made inaccessible), it is unavailable for use by client applications, database-defined stored procedures, triggers, and events. Secure feature settings apply to all databases that are running on the selected database server. Secure features are useful when you need to start a database that might contain embedded logic that you are unsure about, such as a virus, or if you want to lock down a database server or database hosted by a third-party vendor. The `-sf` database server option allows you to specify which features you want to secure for databases running on the database server.

Secure Feature Keys

A `system secure feature key` is created by specifying the `-sk` database server option when creating the database server. Use the `sa_server_option` system procedure to alter whether features are secured or unsecured once the database server is running.

Once you have created a system secure feature key, you can create `customized secure feature keys` that are assigned to a specific users, limiting users' access to only the features secured by the administrator for that key.

Customized secure feature keys are managed by select system procedures.

In this section:

1.12.1.1 Creating secure feature keys

Control the database features available to users, by using the secure features database server option (-sf) to specify the features that users are prevented from accessing on the database server.

Prerequisites

You must have the SERVER OPERATOR system privilege and have access to the manage_keys feature.

Context

Secure feature settings apply to all databases running on a database server.

The secure features option (-sf) controls the availability of such features as:

- Server-side backups
- External stored procedures
- Remote data access
- Web services

The -sk option specifies a system secure feature key that manages access to secure features for a database server. To alter the list of secured features once the database server is running, use the sa_server_option system procedure. To alter a customized secure feature key once the database server is running, use the sp_alter_secure_feature_key system procedure.

The sp_create_secure_feature_key system procedure creates a customized secure feature key.

Procedure

1. At a command prompt, start the database server using the -sf and -sk options.

   For example, the following command starts the database server and secures all features. The command also includes a key that can be used later to allow access to secured features for a connection.

   ```
   start_iq -n secure_server -sf all -sk someSystemKey c:\mydemo.db
   ```

2. Connect to the database server:

   ```
   dbisql -c "UID=DBA;PWD=sql;Host=myhost;Server=secure_server;DBN=mydemo"
   ```

3. Call the sp_use_secure_feature_key system procedure to specify the secure feature key for the connection. In this case, the secure feature key is the same as that specified by the -sk option:

   ```
   CALL sp_use_secure_feature_key ( 'system' , 'someSystemKey' );
   ```

4. Change the set of secured features on the server by using the sa_server_option system procedure.

   For example:

   ```
   CALL sa_server_option( 'all', '-remote_data_access' );
   ```

5. Create a customized secure feature key for a specific user.

   For example, create a customized secure feature key for Bob that allows him to send emails:

   ```
   CALL sp_create_secure_feature_key ( 'bobsKey' , 'anotherAuthKey' ,
   'sa_send_email' );
   ```

   After logging into the database, Bob must run the following command to send emails:

   ```
   CALL sp_use_secure_feature_key ( 'bobsKey' , 'anotherAuthKey' );
   ```

Results

There is now a system secure feature for the database server, as well as a customized secure feature that has been assigned to a specific user.

Users of databases running on the database server secure_server are prevented from accessing all secured features except the remote_data_access feature. The user Bob, however, also has access to the sa_send_email feature.

Related Information

-sf database server option [page 367]
-sk database server option [page 366]
sp_alter_secure_feature_key System Procedure [page 443]
sp_create_secure_feature_key System Procedure [page 444]
sp_drop_secure_feature_key System Procedure [page 445]
sp_list_secure_feature_keys System Procedure [page 445]
sp_use_secure_feature_key System Procedure [page 446]

2 External Authentication

SAP IQ supports LDAP and Kerberos external authentication methods.

In this section:

2.1 LDAP User Authentication with SAP IQ

You can integrate SAP IQ into any existing enterprise-wide directory access framework based on Lightweight Directory Access Protocol (LDAP), a widely accepted international standard.

Integration of SAP IQ with LDAP user authentication supports:

- Authentication using searched distinguished name (DN)
- Failover to a secondary LDAP server for high availability
- Automatic failback to previously failed servers
- Integration with OpenLDAP third-party libraries
- Secure communication with LDAP servers
- Efficient design for frequent, short-lived connections
- Extensibility to multiple domains and multiple LDAP servers

In this section:

To support failover functionality, you can create a primary and a secondary LDAP server configuration object.

Enabling LDAP User Authentication [page 195]
Configure LDAP user authentication with SAP IQ. Once configuration is complete verify that users can log on using LDAP user authentication.

Managing the LDAP Server Configuration Object with SAP IQ [page 205]
Management includes the creation, modification and option maintenance of the LDAP server configuration object to facilitate LDAP user authentication.

Managing LDAP User Authentication Login Policy Options [page 221]
There are several login policy options specific to LDAP user authentication. These options must be defined in any login policy (including root) assigned to a user using LDAP user authentication.

Manage Users and Passwords with LDAP User Authentication [page 225]
To log in to SAP IQ using LDAP user authentication, each user must have an active user ID and password on the external LDAP server as well as an active user ID on the SAP IQ server.

Displaying Current Status Information for a User [page 226]
Run the `sa_get_user_status` stored procedure to generate a report about the current status of a user.

Displaying Current State for an LDAP Server Configuration Object [page 226]
Run the `sa_get_ldapserver_status` stored procedure to generate a report on the current state of an LDAP server configuration object.

2.1.1 License Requirements for LDAP User Authentication

The Advanced Security Option (IQ_SECURITY) protects your environment against unauthorized access, and is required to allow LDAP user authentication with SAP IQ.

2.1.2 About the LDAP Server Configuration Object

SAP IQ uses a configuration object called LDAP server to allow LDAP user authentication.

Despite its name, the LDAP server is a configuration object that resides on the SAP IQ server, rather than an actual server. Its sole function is to provide a connection to a physical LDAP server to allow LDAP user authentication. Any configuration of the LDAP server configuration object applies only to the SAP IQ side of the LDAP user authentication equation. LDAP server configuration object configuration settings are never written to the physical LDAP server.

> i Note
>
> For the purposes of clarity in this documentation, LDAP server configuration object refers to the SAP IQ internal configuration object. LDAP server refers to the external entity.

2.1.3 Failover Capabilities When Using LDAP User Authentication

To support failover functionality, you can create a primary and a secondary LDAP server configuration object.

Each LDAP server configuration object connects to a single LDAP server and can be designated as a primary or secondary server. In the event the designated primary LDAP server configuration object is cannot connect to the LDAP server, the designated secondary LDAP server configuration object is used for user authentication. You can manually manage fail over and fail back using with SQL statements or be performed automatically by SAP IQ when it detects a change is appropriate.

Define primary and secondary LDAP server configuration objects in the login policy. For failover to occur, you must define both a primary and a secondary LDAP server configuration object. If only a primary LDAP server configuration object is defined in a login policy, failover does not occur. If a secondary LDAP server configuration object is defined with no primary LDAP server configuration object, the secondary LDAP server configuration object behaves as the primary LDAP server configuration object, and failover does not occur.

When designating the secondary LDAP server configuration object, you must configure the LDAP server configuration object to connect to the correct failover LDAP server. In the event of a failover, if the secondary LDAP server configuration object cannot connect to the secondary LDAP server, LDAP user authentication in SAP IQ will be unavailable.

2.1.4 Enabling LDAP User Authentication

Configure LDAP user authentication with SAP IQ. Once configuration is complete verify that users can log on using LDAP user authentication.

1. Configuring LDAP User Authentication as a Login Method [page 196]
   To enable LDAP user authentication, you must add the value `LDAPUA` to the LOGIN_MODE database option.
2. Creating an LDAP Server Configuration Object [page 196]
   Create a new LDAP server configuration object to allow LDAP user authentication.
3. Validating an LDAP Server Configuration Object [page 198]
   Validate the attribute of a new or existing LDAP server configuration object.
4. Managing LDAP User Authentication Login Policy Options [page 201]
   There are several login policy options specific to LDAP user authentication. These options must be defined in any login policy (including root) assigned to a user using LDAP user authentication.
5. Displaying Current State for an LDAP Server Configuration Object [page 205]
   Run the `sa_get_ldapserver_status` stored procedure to generate a report on the current state of an LDAP server configuration object.

2.1.4.1 Configuring LDAP User Authentication as a Login Method

To enable LDAP user authentication, you must add the value `LDAPUA` to the LOGIN_MODE database option.

Prerequisites

Requires the SET ANY SECURITY OPTION system privilege.

Context

Once set, LDAP user authentication is immediately available.

Procedure

To add the LDAPUA value to the LOGIN_MODE option, execute:

```
SET OPTION PUBLIC.login_mode = LDAPUA
```

Task overview: Enabling LDAP User Authentication [page 195]

Next task: Creating an LDAP Server Configuration Object [page 196]

2.1.4.2 Creating an LDAP Server Configuration Object

Create a new LDAP server configuration object to allow LDAP user authentication.

Prerequisites

Requires the MANAGE ANY LDAP SERVER system privilege.

Context

The LDAP server configuration object provides a connection between SAP IQ and a physical LDAP server. If you are using multiple LDAP servers, particularly for failover, set up a separate LDAP server configuration object for each LDAP server. The parameters of the LDAP server configuration object are stored in the `ISYSLDAPSERVER` (system view SYSLDAPSERVER) system table. To automatically activate the connection to the LDAP server upon creation, use the WITH ACTIVATE clause.

Procedure

1. Identify the values for the applicable SEARCH DN attributes to be defined for the new LDAP server configuration object.

   Table 2: SEARCH DN Attributes

Attribute	Valid Values
URL	Specify the host (by name or by IP address), port number, and search to be performed to lookup the DN for a given user ID or enter NULL. Note See *Syntax and Parameters for the LDAP Server Configuration Object URL* for supported syntax.
ACCESS ACCOUNT	The distinguished name for a user connecting to the external LDAP server.
IDENTIFIED BY	The password associated with the ACCESS ACCOUNT distinguished name.
IDENTIFIED BY ENCRYPTED	The encrypted password associated with the ACCESS ACCOUNT distinguished name.

2. Identify the values for the applicable LDAPUA server attributes for the new LDAP server configuration object.

   Table 3: LDAPUA Attributes

Attribute	Valid Values
SEARCH DN	All attributes defined from SEARCH DN Attributes (see step 1).
AUTHENTICATION URL	Specify the host (by name or by IP address), port number, and search to be performed to lookup the DN for a given user ID or enter NULL. Note See *Syntax and Parameters for the LDAP Server Configuration Object URL* for supported syntax.
CONNECTION TIMEOUT	Specifies the connection timeout value for both DN searches and authentication between SAP IQ and the external LDAP server. Specified in milliseconds, the default value is 10 seconds.

Attribute	Valid Values
CONNECTION RETRIES	Specifies the number of retries on connections from SAP IQ to the LDAP server for both DN searches and authentication. The valid range of values is 1 – 60, with a default value of 3.
TLS	Defines whether the TLS or Secure LDAP protocol is used for connections to the LDAP server both for DN searches and authentication. The valid settings are ON and OFF (default). i Note See *Enabling Secure LDAP* and *Setting the TLS Connection Trusted Relationship*.

3. Execute the `CREATE LDAP SERVER` command, specifying the applicable attributes and clauses. For example:

Example

```
CREATE LDAP SERVER secure_primary
SEARCH DN
     URL 'ldaps://my_LDAPserver:636/dc=MyCompany,dc=com??sub?cn=*'
     ACCESS ACCOUNT 'cn=myadmin, cn=Users, dc=mycompany, dc=com'
     IDENTIFIED BY 'Secret99Password'
AUTHENTICATION URL 'ldaps://my_LDAPserver:636/'
CONNECTION TIMEOUT 3000
CONNECTION RETRIES 3
TLS OFF
WITH ACTIVATE
```

Task overview: Enabling LDAP User Authentication [page 195]

Previous task: Configuring LDAP User Authentication as a Login Method [page 196]

Next task: Validating an LDAP Server Configuration Object [page 198]

2.1.4.3 Validating an LDAP Server Configuration Object

Validate the attribute of a new or existing LDAP server configuration object.

Prerequisites

Requires the MANAGE ANY LDAP SERVER system privilege.

Context

The `VALIDATE LDAP SERVER` command is useful for an administrator when setting up a new LDAP server configuration object or when diagnosing connection issues between SAP IQ and the LDAP server. Any connection established by the VALIDATE LDAP SERVER statement is temporary and closed at the end of the execution of the statement.

To validate the existence of the user on the LDAP server, include the `CHECK` clause. Specify the userID and the `<user-dn-string>` to be compared.

Procedure

1. Identify the SEARCH DN attributes of the LDAP server configuration object to be validated.

Table 4: SEARCH DN Attributes

Attribute	Valid Values
URL	Specify the host (by name or by IP address), port number, and search to be performed to lookup the DN for a given user ID or enter NULL. i Note See *Syntax and Parameters for the LDAP Server Configuration Object URL* for supported syntax.
ACCESS ACCOUNT	The distinguished name for a user connecting to the external LDAP server.
IDENTIFIED BY	The password associated with the ACCESS ACCOUNT distinguished name.
IDENTIFIED BY ENCRYPTED	The encrypted password associated with the ACCESS ACCOUNT distinguished name.

2. Identify the LDAPUA attributes of the LDAP server configuration object to be validated.

Table 5: LDAPUA Attributes

Attribute	Valid Values
SEARCH DN	All attributes defined from SEARCH DN Attributes (see step 1).
AUTHENTICATION URL	Specify the host (by name or by IP address), port number, and search to be performed to lookup the DN for a given user ID or enter NULL. i Note See *Syntax and Parameters for the LDAP Server Configuration Object URL* for supported syntax.
CONNECTION TIMEOUT	Specifies the connection timeout value for both DN searches and authentication between SAP IQ and the external LDAP server. Specified in milliseconds, the default value is 10 seconds.

Attribute	Valid Values
CONNECTION RETRIES	Specifies the number of retries on connections from SAP IQ to the LDAP server for both DN searches and authentication. The valid range of values is 1 – 60, with a default value of 3.
TLS	Defines whether the TLS or Secure LDAP protocol is used for connections to the LDAP server both for DN searches and authentication. The valid settings are ON and OFF (default). i Note See *Enabling Secure LDAP* and *Setting the TLS Connection Trusted Relationship*.

3. Execute the `VALIDATE LDAP SERVER` command with the applicable attributes.

Example

For example, assume the LDAP server configuration object named `apps_primary` was created as follows and the `SET OPTION PUBLIC.login_mode` is set to `‘Standard,LDAPUA’`:

```
CREATE LDAP SERVER apps_primary
SEARCH DN
    URL 'ldap://my_LDAPserver:389/dc=MyCompany,dc=com??sub?cn=*'
    ACCESS ACCOUNT 'cn=myadmin, cn=Users, dc=mycompany, dc=com'
    IDENTIFIED BY 'Secret99Password'
AUTHENTICATION URL 'ldap://my_LDAPserver:389/'
CONNECTION TIMEOUT 3000
WITH ACTIVATE
```

This statement validates the existence of a userID `myusername` by comparing it to the expected user distinguished name (enclosed in quotation marks) on the LDAP server configuration object name `apps_primary` using the optional CHECK clause:

```
VALIDATE LDAP SERVER apps_primary
CHECK myusername 'cn=myusername,cn=Users,dc=mycompany,dc=com'
```

Task overview: Enabling LDAP User Authentication [page 195]

Previous task: Creating an LDAP Server Configuration Object [page 196]

Next: Managing LDAP User Authentication Login Policy Options [page 201]

2.1.4.4 Managing LDAP User Authentication Login Policy Options

There are several login policy options specific to LDAP user authentication. These options must be defined in any login policy (including root) assigned to a user using LDAP user authentication.

You can define the options that are specific to LDAP server database objects when initially creating a login policy, or you can add them to existing policies, including the root login policy.

Requires the MANAGE ANY LOGIN POLICY system privilege to define login policy options.

In this section:

Parent topic: Enabling LDAP User Authentication [page 195]

Previous task: Validating an LDAP Server Configuration Object [page 198]

Next: Displaying Current State for an LDAP Server Configuration Object [page 205]

2.1.4.4.1 Modifying the Root Login Policy

You can modify the option values for the root login policy, but you cannot drop the policy.

Prerequisites

The MANAGE ANY LOGIN POLICY system privilege.

Context

Each new database is created with a default login policy, called the root policy. When you create a user account without specifying a login policy, the user becomes part of the root login policy.

Procedure

To modify the options of the root login policy, execute:

```
ALTER LOGIN POLICY ROOT <{login_policy_options}>
```

Task overview: Managing LDAP User Authentication Login Policy Options [page 201]

Related Information

Modifying an Existing Login Policy [page 202]
Creating a New Login Policy [page 203]
Assigning a Login Policy to an Existing User [page 204]

2.1.4.4.2 Modifying an Existing Login Policy

Modify options within an existing login policy.

Prerequisites

The MANAGE ANY LOGIN POLICY system privilege.

Procedure

To alter the options of an existing login policy, execute:

```
ALTER LOGIN POLICY <policy-name> <{login_policy_options}>
```

Example

This statement alters the LOCKED and MAX_CONNECTIONS options on the `Test1` login policy:

```
ALTER LOGIN POLICY Test1
locked=on
max_connections=5
```

Task overview: Managing LDAP User Authentication Login Policy Options [page 201]

Related Information

Modifying the Root Login Policy [page 201]
Creating a New Login Policy [page 203]
Assigning a Login Policy to an Existing User [page 204]

2.1.4.4.3 Creating a New Login Policy

Any options that are not explicitly set when creating a login policy inherit their values from the root login policy.

Prerequisites

The MANAGE ANY LOGIN POLICY system privilege.

Context

Login policy names must be unique. You see an error message if the name of the login policy you are adding already exists.

Procedure

To create a new login policy, execute:

```
CREATE LOGIN POLICY <policy_name> <{login_policy_options}>
```

Example

This statement creates the `Test1` login policy with PASSWORD_LIVE_TIME option set to 60 days:

```
CREATE LOGIN POLICY Test1
password_life_time=60
```

Task overview: Managing LDAP User Authentication Login Policy Options [page 201]

Related Information

Modifying the Root Login Policy [page 201]
Modifying an Existing Login Policy [page 202]
Assigning a Login Policy to an Existing User [page 204]

2.1.4.4.4 Assigning a Login Policy to an Existing User

Assign a login policy to an existing SAP IQ user.

Prerequisites

The MANAGE ANY LOGIN POLICY system privilege.

Procedure

1. Execute:

   ```
   ALTER USER <userID>
   LOGIN POLICY <policy_name>
   ```

2. Have the user log out and back in to apply the new login policy.

Task overview: Managing LDAP User Authentication Login Policy Options [page 201]

Related Information

Modifying the Root Login Policy [page 201]
Modifying an Existing Login Policy [page 202]
Creating a New Login Policy [page 203]

2.1.4.5 Displaying Current State for an LDAP Server Configuration Object

Run the `sa_get_ldapserver_status` stored procedure to generate a report on the current state of an LDAP server configuration object.

Status information includes the LDAP server configuration object name, object identifier, current state, and the date and time of the last state change. A properly configured and running LDAP server configuration object has a state of READY or ACTIVE.

No system privilege is required to run this stored procedure.

Parent topic: Enabling LDAP User Authentication [page 195]

Previous: Managing LDAP User Authentication Login Policy Options [page 201]

2.1.5 Managing the LDAP Server Configuration Object with SAP IQ

Management includes the creation, modification and option maintenance of the LDAP server configuration object to facilitate LDAP user authentication.

In this section:

Configuring LDAP User Authentication as a Login Method [page 206]
To enable LDAP user authentication, you must add the value `LDAPUA` to the LOGIN_MODE database option.

Setting the TLS Connection Trusted Relationship [page 208]
Define the location and file name that contains the trusted relationship to be used for the Transport Layer Security (TLS) connections to the external LDAP server for user authentication.

Creating an LDAP Server Configuration Object [page 209]
Create a new LDAP server configuration object to allow LDAP user authentication.

Validating an LDAP Server Configuration Object [page 211]
Validate the attribute of a new or existing LDAP server configuration object.

Activating an LDAP Server Configuration Object [page 213]
Activate an LDAP server configuration object by setting the connection state to READY. This enables LDAP user authentication.

Editing LDAP Server Configuration Object Attributes [page 214]
Modify the existing attributes on an LDAP server. Any changes to the attributes are applied on subsequent connections. Any connection already open when the change is applied does not immediately reflect the change.

Refreshing an LDAP Server Configuration Object [page 216]
Reinitialize the LDAP server. The command fails if the connection state of the LDAP server is not in an ACTIVE or READY state.

Suspending an LDAP Server Configuration Object [page 217]
: Put an LDAP server into maintenance mode. All connections to the LDAP server are closed and LDAP user authentication is no longer available.

Deleting an LDAP Server Configuration Object [page 218]
: Delete an LDAP server configuration object that is not in a READY or ACTIVE state.

LDAP Server Configuration Object States [page 219]
: List of possible states of an LDAP server configuration object.

Enabling Secure LDAP [page 219]
: Secure LDAP uses TLS certificate authentication to provide protection against spoofing.

Syntax and Parameters for the LDAP Server Configuration Object URL [page 220]
: The URL identifies the host (by name or by IP address), port number, and search to be performed when executing a secure distinguished name (DN) lookup to the LDAP server.

2.1.5.1 Configuring LDAP User Authentication as a Login Method

To enable LDAP user authentication, you must add the value `LDAPUA` to the LOGIN_MODE database option.

Prerequisites

Requires the SET ANY SECURITY OPTION system privilege.

Context

Once set, LDAP user authentication is immediately available.

Procedure

To add the LDAPUA value to the LOGIN_MODE option, execute:

```
SET OPTION PUBLIC.login_mode = LDAPUA
```

In this section:

Allowing Standard Authentication in an LDAP User Authentication Only Environment [page 207]
: Allow select users to authenticate using standard authentication in an environment that supports only LDAP user authentication.

Related Information

LOGIN_MODE Option [page 356]

2.1.5.1.1 Allowing Standard Authentication in an LDAP User Authentication Only Environment

Allow select users to authenticate using standard authentication in an environment that supports only LDAP user authentication.

Context

If LDAP user authentication is the only authentication method allowed to access the SAP IQ database, these circumstances may create a scenario in which no user is permitted to log on:

- Of no login policy exists with LDAP user authentication enabled;
- If no users are assigned to a login policy with LDAP user authorization enabled; or
- If all user accounts assigned to a login policy with LDAP user authentication are locked.

You may not be able to prevent this scenario; however, there is a method that allows a select number of users to log in to SAP IQ database using standard authentication. This method is intended as a temporary solution when LOGIN_MODE configuration prevents all users from connecting to the database.

When granting the select users access using standard authentication, ensure that at least one of those users has the SET ANY SECURITY OPTION or MANAGE ANY LOGIN POLICY system privileges to allow them to permanently resolve the issue. Depending on the underlying cause of the inability of any users to log in using LDAP user authentication, one or both of these system privileges might be required to permanently resolve the issue. You can specify a maximum of five user IDs, separated by semicolons, and enclosed in double quotation marks.

Grant standard authentication access only after the lockdown problem has occurred; you need not set it in advance. It does not need to be set in advance. To allow select users to log in using standard authentication, execute the `start_iq` utility with the `-al <user-id-list>` command line switch. Once granted, at the credentials prompt, the user enters his or her standard authentication user name and password.

Include the `-al` switch at either the server or database level. At the server level, the `-al` switch remains in effect until the next time the server is restarted. At the database level, the `-al` switch remains in effect until the next time the database is stopped and restarted.

Procedure

To allow standard authentication, execute one of these commands:

Level	Statement
Server	`start_iq -al <"user1,user2,user3" server_name.cfg database-name.db >`
Database	`start_iq <servername.cfg database_name.db> -al <"user1,user2,user3">`

Example

This example assumes that login_mode is set to "LDAPUA". This command allows users `Alice`, `Bob`, and `Carol` to authenticate using standard authentication on `database1` on `server1`:

```
start_iq -al  "alice;bob;carol" server1.cfg database1.db
```

Related Information

-al database server option [page 359]
-al database option [page 359]

2.1.5.2 Setting the TLS Connection Trusted Relationship

Define the location and file name that contains the trusted relationship to be used for the Transport Layer Security (TLS) connections to the external LDAP server for user authentication.

Prerequisites

Requires the SET ANY SECURITY OPTION system privilege.

Context

During LDAP user authentication, SAP IQ acts as a client to the LDAP server, and must have access to the file that contains the name of the certificate authority (CA) that signed the TLS certificate. The path and file name to the CA are stored in the public-only TRUSTED_CERTIFICATES_FILE database security option. By default, this option is set to NULL (disabled), meaning that no outbound connections can be started because there are no trusted CA. Once set, this value takes effect immediately.

The list of trusted CAs that sign server certificates may be shared in a location in a Windows environment on the local C: drive for all SAP applications on that machine.

Procedure

To set the TRUSTED_CERTIFICATES_FILE database security option, execute:

```
SET OPTION PUBLIC.TRUSTED_CERTIFICATES_FILE = <‘path/filename’>
```

Example

This example sets the path to the trusted certificates file to `C:\sybase\shared`, in a file called `\trusted.txt`:

```
SET OPTION PUBLIC.TRUSTED_CERTIFICATES_FILE = <‘C:\sybase\shared\trusted.txt’>
```

Related Information

TRUSTED_CERTIFICATES_FILE Option [page 358]

2.1.5.3 Creating an LDAP Server Configuration Object

Create a new LDAP server configuration object to allow LDAP user authentication.

Prerequisites

Requires the MANAGE ANY LDAP SERVER system privilege.

Context

The LDAP server configuration object provides a connection between SAP IQ and a physical LDAP server. If you are using multiple LDAP servers, particularly for failover, set up a separate LDAP server configuration object for each LDAP server. The parameters of the LDAP server configuration object are stored in the `ISYSLDAPSERVER` (system view SYSLDAPSERVER) system table. To automatically activate the connection to the LDAP server upon creation, use the WITH ACTIVATE clause.

Procedure

1. Identify the values for the applicable SEARCH DN attributes to be defined for the new LDAP server configuration object.

   Table 6: SEARCH DN Attributes

Attribute	Valid Values
URL	Specify the host (by name or by IP address), port number, and search to be performed to lookup the DN for a given user ID or enter NULL. i Note See *Syntax and Parameters for the LDAP Server Configuration Object URL* for supported syntax.
ACCESS ACCOUNT	The distinguished name for a user connecting to the external LDAP server.
IDENTIFIED BY	The password associated with the ACCESS ACCOUNT distinguished name.
IDENTIFIED BY ENCRYPTED	The encrypted password associated with the ACCESS ACCOUNT distinguished name.

2. Identify the values for the applicable LDAPUA server attributes for the new LDAP server configuration object.

   Table 7: LDAPUA Attributes

Attribute	Valid Values
SEARCH DN	All attributes defined from SEARCH DN Attributes (see step 1).
AUTHENTICATION URL	Specify the host (by name or by IP address), port number, and search to be performed to lookup the DN for a given user ID or enter NULL. i Note See *Syntax and Parameters for the LDAP Server Configuration Object URL* for supported syntax.
CONNECTION TIMEOUT	Specifies the connection timeout value for both DN searches and authentication between SAP IQ and the external LDAP server. Specified in milliseconds, the default value is 10 seconds.
CONNECTION RETRIES	Specifies the number of retries on connections from SAP IQ to the LDAP server for both DN searches and authentication. The valid range of values is 1 – 60, with a default value of 3.
TLS	Defines whether the TLS or Secure LDAP protocol is used for connections to the LDAP server both for DN searches and authentication. The valid settings are ON and OFF (default). i Note See *Enabling Secure LDAP* and *Setting the TLS Connection Trusted Relationship*.

3. Execute the `CREATE LDAP SERVER` command, specifying the applicable attributes and clauses. For example:

Example

```
CREATE LDAP SERVER secure_primary
SEARCH DN
     URL 'ldaps://my_LDAPserver:636/dc=MyCompany,dc=com??sub?cn=*'
     ACCESS ACCOUNT 'cn=myadmin, cn=Users, dc=mycompany, dc=com'
     IDENTIFIED BY 'Secret99Password'
AUTHENTICATION URL 'ldaps://my_LDAPserver:636/'
CONNECTION TIMEOUT 3000
CONNECTION RETRIES 3
TLS OFF
WITH ACTIVATE
```

Related Information

Syntax and Parameters for the LDAP Server Configuration Object URL [page 220]
Enabling Secure LDAP [page 219]
CREATE LDAP SERVER Statement [page 298]
Editing LDAP Server Configuration Object Attributes [page 214]
Setting the TLS Connection Trusted Relationship [page 208]

2.1.5.4 Validating an LDAP Server Configuration Object

Validate the attribute of a new or existing LDAP server configuration object.

Prerequisites

Requires the MANAGE ANY LDAP SERVER system privilege.

Context

The `VALIDATE LDAP SERVER` command is useful for an administrator when setting up a new LDAP server configuration object or when diagnosing connection issues between SAP IQ and the LDAP server. Any connection established by the VALIDATE LDAP SERVER statement is temporary and closed at the end of the execution of the statement.

To validate the existence of the user on the LDAP server, include the `CHECK` clause. Specify the userID and the `<user-dn-string>` to be compared.

Procedure

1. Identify the SEARCH DN attributes of the LDAP server configuration object to be validated.

Table 8: SEARCH DN Attributes

Attribute	Valid Values
URL	Specify the host (by name or by IP address), port number, and search to be performed to lookup the DN for a given user ID or enter NULL. i Note See *Syntax and Parameters for the LDAP Server Configuration Object URL* for supported syntax.
ACCESS ACCOUNT	The distinguished name for a user connecting to the external LDAP server.
IDENTIFIED BY	The password associated with the ACCESS ACCOUNT distinguished name.
IDENTIFIED BY ENCRYPTED	The encrypted password associated with the ACCESS ACCOUNT distinguished name.

2. Identify the LDAPUA attributes of the LDAP server configuration object to be validated.

Table 9: LDAPUA Attributes

Attribute	Valid Values
SEARCH DN	All attributes defined from SEARCH DN Attributes (see step 1).
AUTHENTICATION URL	Specify the host (by name or by IP address), port number, and search to be performed to lookup the DN for a given user ID or enter NULL. i Note See *Syntax and Parameters for the LDAP Server Configuration Object URL* for supported syntax.
CONNECTION TIMEOUT	Specifies the connection timeout value for both DN searches and authentication between SAP IQ and the external LDAP server. Specified in milliseconds, the default value is 10 seconds.
CONNECTION RETRIES	Specifies the number of retries on connections from SAP IQ to the LDAP server for both DN searches and authentication. The valid range of values is 1 – 60, with a default value of 3.
TLS	Defines whether the TLS or Secure LDAP protocol is used for connections to the LDAP server both for DN searches and authentication. The valid settings are ON and OFF (default). i Note See *Enabling Secure LDAP* and *Setting the TLS Connection Trusted Relationship*.

3. Execute the `VALIDATE LDAP SERVER` command with the applicable attributes.

Example

For example, assume the LDAP server configuration object named `apps_primary` was created as follows and the `SET OPTION PUBLIC.login_mode` is set to `‘Standard,LDAPUA’`:

```
CREATE LDAP SERVER apps_primary
SEARCH DN
    URL 'ldap://my_LDAPserver:389/dc=MyCompany,dc=com??sub?cn=*'
    ACCESS ACCOUNT 'cn=myadmin, cn=Users, dc=mycompany, dc=com'
    IDENTIFIED BY 'Secret99Password'
AUTHENTICATION URL 'ldap://my_LDAPserver:389/'
CONNECTION TIMEOUT 3000
WITH ACTIVATE
```

This statement validates the existence of a userID `myusername` by comparing it to the expected user distinguished name (enclosed in quotation marks) on the LDAP server configuration object name `apps_primary` using the optional CHECK clause:

```
VALIDATE LDAP SERVER apps_primary
CHECK myusername 'cn=myusername,cn=Users,dc=mycompany,dc=com'
```

Related Information

Enabling Secure LDAP [page 219]
Syntax and Parameters for the LDAP Server Configuration Object URL [page 220]
VALIDATE LDAP SERVER Statement [page 353]
Editing LDAP Server Configuration Object Attributes [page 214]
Setting the TLS Connection Trusted Relationship [page 208]

2.1.5.5 Activating an LDAP Server Configuration Object

Activate an LDAP server configuration object by setting the connection state to READY. This enables LDAP user authentication.

Prerequisites

Requires the MANAGE ANY LDAP SERVER system privilege.

Context

LDAP server configuration object attribute values are read from the ISYSLDAPSERVER system table and applied to new connections to the LDAP server and incoming authentication requests to the SAP IQ server. Upon successful authentication of a user, the connection state to the LDAP server changes to ACTIVE.

Procedure

To activate an LDAP server configuration object, execute:

```
ALTER LDAP SERVER <LDAP_server_name>
WITH ACTIVATE
```

Related Information

ALTER LDAP SERVER Statement [page 282]
LDAP Server Configuration Object States [page 219]

2.1.5.6 Editing LDAP Server Configuration Object Attributes

Modify the existing attributes on an LDAP server. Any changes to the attributes are applied on subsequent connections. Any connection already open when the change is applied does not immediately reflect the change.

Prerequisites

Requires the MANAGE ANY LDAP SERVER system privilege.

Procedure

1. Identify the existing SEARCH DN attributes to be modified.

Table 10: SEARCH DN Attributes

Attribute	Valid Values
URL	Specify the host (by name or by IP address), port number, and search to be performed to lookup the DN for a given user ID or enter NULL. i Note See *Syntax and Parameters for the LDAP Server Configuration Object URL* for supported syntax.
ACCESS ACCOUNT	The distinguished name for a user connecting to the external LDAP server.
IDENTIFIED BY	The password associated with the ACCESS ACCOUNT distinguished name.
IDENTIFIED BY ENCRYPTED	The encrypted password associated with the ACCESS ACCOUNT distinguished name.

2. Identify the existing LDAPUA attributes to be modified.

Table 11: LDAPUA Attributes

Attribute	Valid Values
SEARCH DN	All attributes defined from SEARCH DN Attributes (see step 1).
AUTHENTICATION URL	Specify the host (by name or by IP address), port number, and search to be performed to lookup the DN for a given user ID or enter NULL. i Note See *Syntax and Parameters for the LDAP Server Configuration Object URL* for supported syntax.
CONNECTION TIMEOUT	Specifies the connection timeout value for both DN searches and authentication between SAP IQ and the external LDAP server. Specified in milliseconds, the default value is 10 seconds.
CONNECTION RETRIES	Specifies the number of retries on connections from SAP IQ to the LDAP server for both DN searches and authentication. The valid range of values is 1 – 60, with a default value of 3.
TLS	Defines whether the TLS or Secure LDAP protocol is used for connections to the LDAP server both for DN searches and authentication. The valid settings are ON and OFF (default). i Note See *Enabling Secure LDAP* and *Setting the TLS Connection Trusted Relationship*.

3. Identify the server clauses to be used.

Clause	Description
WITH SUSPEND	Puts the LDAP server into maintenance mode
WITH ACTIVATE	Puts the LDAP server in a READY state and enables LDAP authentication

Clause	Description
WITH REFRESH	Reinitializes LDAP user authentication

4. Execute the ALTER LDAP SERVER command with the applicable parameters and clauses, for example:

Example

```
ALTER LDAP SERVER apps_primary
AUTHENTICATION URL 'ldap://my_LDAPserver:1066/'
CONNECTION RETRIES 10
WITH ACTIVATE
```

Related Information

Syntax and Parameters for the LDAP Server Configuration Object URL [page 220]

Enabling Secure LDAP [page 219]

ALTER LDAP SERVER Statement [page 282]

Setting the TLS Connection Trusted Relationship [page 208]

Validating an LDAP Server Configuration Object [page 211]

2.1.5.7 Refreshing an LDAP Server Configuration Object

Reinitialize the LDAP server. The command fails if the connection state of the LDAP server is not in an ACTIVE or READY state.

Prerequisites

Requires the MANAGE ANY LDAP SERVER system privilege.

Context

When refreshing an LDAP server, all connections to the LDAP server are closed and the option values on the LDAP server are reread from the ISYSLDAPSERVER system table. The values are then applied to all new connections to the LDAP server and all incoming user authentication requests to the SAP IQ server. Execution of the REFRESH command does not change the connection state of the LDAP server, nor does it change any existing connections from a client to the SAP IQ server.

To ensure that any changes are used when a user next authenticates, it is recommended that you refresh the LDAP server after making any changes to the TRUSTED_CERTIFICATES_FILE database option or to the contents of the file specified by the TRUSTED_CERTIFICATES_FILE database option.

Procedure

To refresh the LDAP server, execute:

```
ALTER LDAP SERVER <LDAP_server_name>
WITH REFRESH
```

Related Information

ALTER LDAP SERVER Statement [page 282]
LDAP Server Configuration Object States [page 219]

2.1.5.8 Suspending an LDAP Server Configuration Object

Put an LDAP server into maintenance mode. All connections to the LDAP server are closed and LDAP user authentication is no longer available.

Prerequisites

Requires the MANAGE ANY LDAP SERVER system privilege.

Procedure

To suspend an LDAP server, execute:

```
ALTER LDAP SERVER <LDAP_server_name>
WITH SUSPEND
```

Related Information

ALTER LDAP SERVER Statement [page 282]

LDAP Server Configuration Object States [page 219]

2.1.5.9 Deleting an LDAP Server Configuration Object

Delete an LDAP server configuration object that is not in a READY or ACTIVE state.

Prerequisites

Requires the MANAGE ANY LDAP SERVER system privilege.

Context

The DROP statement fails when it is issued against an LDAP server configuration object that is in a READY or ACTIVE state. The DROP statement also fails if a login policy exists with a reference to the LDAP server configuration object being dropped. To ensure any references to the LDAP server configuration object are removed from all login policies before being dropped, include the WITH DROP ALL REFERENCES clause. To override the server state check and put the database object into maintenance mode regardless of its current state, include the WITH SUSPEND clause when dropping an LDAP server configuration object.

Dropping an LDAP server configuration object removes the named object from the `ISYSLDAPSERVER` system table.

Procedure

To drop an LDAP server configuration object, execute this command, including the applicable clauses:

```
DROP LDAP SERVER <LDAP_Server_name>
WITH SUSPEND
WITH DROP ALL REFERENCES
```

Example

This example drops the LDAP server configuration object named `ldapserver1` regardless of its current state and removes any references to `ldapserver1` in all login policies:

```
DROP LDAP SERVER ldapserver1
WITH DROP ALL REFERENCES
WITH SUSPEND
```

This DROP LDAP SERVER command fails if the LDAP server configuration object named ldapserver2 is referenced in any login policies because the WITH DROP ALL REFERENCES clause is not included:

```
DROP LDAP SERVER ldapserver1
WITH SUSPEND
```

Related Information

DROP LDAP SERVER Statement [page 311]

LDAP Server Configuration Object States [page 219]

2.1.5.10 LDAP Server Configuration Object States

List of possible states of an LDAP server configuration object.

The state of an LDAP server configuration object is maintained persistently on writeable databases in the ISYSLDAPSERVER system table to provide visibility for administrators into LDAP user authentication. If an LDAP server configuration object is restarted, the state at the time of shutdown is retained. This permits maintenance on an LDAP server configuration object to remain in force throughout restarts. With read-only databases, state changes are not stored persistently – they occur only in memory, and are lost when the database is shut down. The connection state is set at start-up using the value from a read-only database, and transient state changes may occur in memory to provide LDAP user authentication.

The possible states of an LDAP server configuration object include:

RESET one or more attributes on the LDAP server configuration object have been entered or modified since last activation.

READY the LDAP server configuration object is ready to accept connections.

ACTIVE the LDAP server configuration object has performed at least one successful LDAP user authentication.

FAILED there is a problem connecting to the LDAP server configuration object.

SUSPENDED the LDAP server configuration object is in maintenance mode, and is unavailable for LDAP user authentication.

2.1.5.11 Enabling Secure LDAP

Secure LDAP uses TLS certificate authentication to provide protection against spoofing.

Use of a TLS certificate provides the client connection to the LDAP server with proof that the server is who it says it is.

Enabling Secure LDAP on an LDAP server configuration object can take one of two forms:

ldaps:// on the LDAP server configuration object, use ldaps:// when defining the SEARCH DN URL or AUTHENTICATION URL attributes and set the TLS attribute to OFF.

TLS parameter on the LDAP server configuration object, use ldap:// when defining the SEARCH DN URL attribute and set the TLS attribute to ON.

> i Note
>
> Current versions of Active Directory (AD), Tivoli, SunONE Oracle DS, and OpenLDAP support both options. Older versions may only support one option. For compatibility with all versions, both options are supported by SAP IQ.

2.1.5.12 Syntax and Parameters for the LDAP Server Configuration Object URL

The URL identifies the host (by name or by IP address), port number, and search to be performed when executing a secure distinguished name (DN) lookup to the LDAP server.

While the syntax of the URL can take one of two forms depending on how the secure connection to the LDAP server is to be made, the underlying parameters of the URL are the same for each form.

ldaps:// on the LDAP server configuration object, use ldaps:// when defining the SEARCH DN URL or AUTHENTICATION URL attributes and set the TLS attribute to OFF.

```
ldapurl::=ldaps://host:[port]/[node]?[attributes]? [base | one | sub]?
[filter]
```

TLS parameter on the LDAP server configuration object, use ldap:// when defining the SEARCH DN URL attribute and set the TLS attribute to ON.

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

Parameter	Description
host	The host name of the LDAP server.
port	The port number of the LDAP server.
node	The node in the object hierarchy at which to start the search.
attributes	A list of attributes returned in the result set. Each LDAP server may support a different attribute based on the schemas used by the LDAP server. However, for each LDAP server, only the first attribute is used and should return the distinguished name (DN) of the user.
base | one | sub	Qualifies the search criteria. base – Specifies a search of the base node. one – Specifies a search of node and one sublevel. sub – Specifies a search of node and all sublevels.
filter	Specifies the attribute or attributes used to search for a database user's distinguished name (DN). The filter can be simple, such as "uid=*," or compound, such as "(uid=*)(ou=group)." The attributes in the filter are dependent on the LDAP server schema. LDAP user authentication replaces each wildcard character (*) with the database user ID when searching for a DN.

The URL is initially defined as one of the server attributes when creating an LDAP server configuration object and can be changed at any time. There are no default values for these parameters. Creating or modifying the LDAP server configuration object requires the MANAGE ANY LDAP SERVER system privilege.

> i Note
>
> Current versions of Active Directory (AD), Tivoli, SunONE Oracle DS, and OpenLDAP support both options. Older versions may only support one option. For compatibility with all versions, both options are supported by SAP IQ.

2.1.6 Managing LDAP User Authentication Login Policy Options

There are several login policy options specific to LDAP user authentication. These options must be defined in any login policy (including root) assigned to a user using LDAP user authentication.

You can define the options that are specific to LDAP server database objects when initially creating a login policy, or you can add them to existing policies, including the root login policy.

Requires the MANAGE ANY LOGIN POLICY system privilege to define login policy options.

In this section:

2.1.6.1 Modifying the Root Login Policy

You can modify the option values for the root login policy, but you cannot drop the policy.

Prerequisites

The MANAGE ANY LOGIN POLICY system privilege.

Context

Each new database is created with a default login policy, called the root policy. When you create a user account without specifying a login policy, the user becomes part of the root login policy.

Procedure

To modify the options of the root login policy, execute:

```
ALTER LOGIN POLICY ROOT <{login_policy_options}>
```

Task overview: Managing LDAP User Authentication Login Policy Options [page 221]

Related Information

Modifying an Existing Login Policy [page 222]
Creating a New Login Policy [page 223]
Assigning a Login Policy to an Existing User [page 225]
Managing LDAP User Authentication Login Policy Options [page 221]
ALTER LOGIN POLICY Statement [page 284]
Login Policy Options [page 302]
LDAP Login Policy Options [page 305]
Multiplex Login Policy Configuration [page 306]

2.1.6.2 Modifying an Existing Login Policy

Modify options within an existing login policy.

Prerequisites

The MANAGE ANY LOGIN POLICY system privilege.

Procedure

To alter the options of an existing login policy, execute:

```
ALTER LOGIN POLICY <policy-name> <{login_policy_options}>
```

Example

This statement alters the LOCKED and MAX_CONNECTIONS options on the `Test1` login policy:

```
ALTER LOGIN POLICY Test1
locked=on
max_connections=5
```

Task overview: Managing LDAP User Authentication Login Policy Options [page 221]

Related Information

Modifying the Root Login Policy [page 221]
Creating a New Login Policy [page 223]
Assigning a Login Policy to an Existing User [page 225]
Managing LDAP User Authentication Login Policy Options [page 221]
ALTER LOGIN POLICY Statement [page 284]
Login Policy Options [page 302]
LDAP Login Policy Options [page 305]
Multiplex Login Policy Configuration [page 306]

2.1.6.3 Creating a New Login Policy

Any options that are not explicitly set when creating a login policy inherit their values from the root login policy.

Prerequisites

The MANAGE ANY LOGIN POLICY system privilege.

Context

Login policy names must be unique. You see an error message if the name of the login policy you are adding already exists.

Procedure

To create a new login policy, execute:

```
CREATE LOGIN POLICY <policy_name> <{login_policy_options}>
```

Example

This statement creates the `Test1` login policy with PASSWORD_LIVE_TIME option set to 60 days:

```
CREATE LOGIN POLICY Test1
password_life_time=60
```

Task overview: Managing LDAP User Authentication Login Policy Options [page 221]

Related Information

Modifying the Root Login Policy [page 221]
Modifying an Existing Login Policy [page 222]
Assigning a Login Policy to an Existing User [page 225]
Managing LDAP User Authentication Login Policy Options [page 221]
CREATE LOGIN POLICY Statement [page 300]
Login Policy Options [page 302]
LDAP Login Policy Options [page 305]
Multiplex Login Policy Configuration [page 306]

2.1.6.4 Assigning a Login Policy to an Existing User

Assign a login policy to an existing SAP IQ user.

Prerequisites

The MANAGE ANY LOGIN POLICY system privilege.

Procedure

1. Execute:

```
ALTER USER <userID>
LOGIN POLICY <policy_name>
```

2. Have the user log out and back in to apply the new login policy.

Task overview: Managing LDAP User Authentication Login Policy Options [page 221]

Related Information

Modifying the Root Login Policy [page 221]
Modifying an Existing Login Policy [page 222]
Creating a New Login Policy [page 223]
ALTER USER Statement [page 294]

2.1.7 Manage Users and Passwords with LDAP User Authentication

To log in to SAP IQ using LDAP user authentication, each user must have an active user ID and password on the external LDAP server as well as an active user ID on the SAP IQ server.

When creating a new user in SAP IQ, though not required, it is recommended that you specify a password to ensure that the new user account is not left unprotected until the first LDAP user authentication login.

The first time a new user logs on or an existing user logs in after a password change, the password in the SAP IQ database is automatically overwritten with the corresponding user password defined on the external LDAP server. Therefore, all maintenance required on SAP IQ passwords for user using LDAP user authentication should always be done on the external LDAP server, not the SAP IQ server.

As a result of this automatic password synchronization, for users granted the ability to use Standard authentication (the password defined in the SAP IQ database), when attempting to log on when using Standard authentication, they should continue to use their LDAP server credentials.

2.1.8 Displaying Current Status Information for a User

Run the `sa_get_user_status` stored procedure to generate a report about the current status of a user.

Information includes connection and failed login information as well as whether the user has been locked out and if so, why. If the user is authenticated using LDAP user authentication, the output includes the user's distinguished name and the date and time that the distinguished name was found.

The MANAGE ANY USER system privilege is required to run this stored procedure. A user without the MANAGE ANY USER system privilege can obtain user information by creating and executing a cover procedure owned by a user with MANAGE ANY USER system privilege.

Related Information

sa_get_user_status system procedure [page 376]

2.1.9 Displaying Current State for an LDAP Server Configuration Object

Run the `sa_get_ldapserver_status` stored procedure to generate a report on the current state of an LDAP server configuration object.

Status information includes the LDAP server configuration object name, object identifier, current state, and the date and time of the last state change. A properly configured and running LDAP server configuration object has a state of READY or ACTIVE.

No system privilege is required to run this stored procedure.

Related Information

sa_get_ldapserver_status System Procedure [page 375]

2.2 Kerberos authentication

The Kerberos login feature allows you to maintain a single user ID and password for database connections, operating system, and network logins.

The Kerberos login is more convenient for users and permits a single security system for database and network security. Its advantages include:

- The user does not need to provide a user ID or password to connect to the database.
- Multiple users can be mapped to a single database user ID.
- The name and password used to log in to Kerberos do not have to match the database user ID and password.

Kerberos is a network authentication protocol that provides strong authentication and encryption using secret-key cryptography. Users already logged in to Kerberos can connect to a database without providing a user ID or password.

Kerberos can be used for authentication. To delegate authentication to Kerberos you must:

- configure the server and database to use Kerberos logins.
- create mapping between the user ID that logs in to the computer or network, and the database user.

> ⚠ Caution
>
> When using Kerberos logins as a single security solution, be sure to inform yourself on the security concern related to copied databases.

SAP IQ does not include the Kerberos software; it must be obtained separately. The following components are included with the Kerberos software:

Kerberos libraries

These are referred to as the Kerberos Client or GSS (Generic Security Services)-API runtime library. These Kerberos libraries implement the well-defined GSS-API. The libraries are required on each client and server computer that intends to use Kerberos. The built-in Windows SSPI interface can be used instead of a third-party Kerberos client library if you are using Active Directory as your KDC.

SSPI can only be used by clients in the Kerberos connection parameter. The database server cannot use SSPI. It needs a supported Kerberos client other than SSPI.

A Kerberos Key Distribution Center (KDC) server

The KDC functions as a storehouse for users and servers. It also verifies the identification of users and servers. The KDC is typically installed on a server computer not intended for applications or user logins.

Kerberos authentication from DBLib, ODBC, OLE DB, and ADO.NET clients, and SAP Open Client and jConnect clients is supported. Kerberos authentication can be used with SAP IQ transport layer security encryption, but Kerberos encryption for network communications is not supported.

Windows uses Kerberos for Windows domains and domain accounts. Active Directory Windows Domain Controllers implement a Kerberos KDC. A third-party Kerberos client or runtime is still required on the database server computer for authentication in this environment, but the Windows client computers can use the built-in Windows SSPI interface instead of a third-party Kerberos client or runtime.

In this section:

Kerberos clients [page 228]
    Kerberos authentication is available on several platforms.

Setting up a Kerberos system to use with SAP IQ [page 229]
    Configure Kerberos authentication to be used with SAP IQ.

Configuring SAP IQ databases to use Kerberos [page 230]
    Configure databases to use Kerberos logins.

Connections from an SAP Open Client or jConnect application [page 232]
    The database server accepts connections from an SAP Open Client or jConnect application.

Connecting using SSPI for Kerberos logins on Windows [page 232]
    Connect using SSPI without a Kerberos client installed on the client computer.

Troubleshooting: Kerberos connections [page 233]
    If you get unexpected errors when attempting to enable or use Kerberos authentication, it is recommended that you enable additional diagnostic messages on the database server and client.

Security: Use login modes to secure the database [page 236]
    There are several measures you can take to secure your database.

2.2.1 Kerberos clients

Kerberos authentication is available on several platforms.

For a list of tested Kerberos clients, see SQL Anywhere Supported Kerberos Clients.

The following table lists the default names and locations of the keytab and GSS-API files used by the supported Kerberos clients.

> **i Note**
>
> SSPI can only be used by clients in the Kerberos connection parameter. The database server cannot use SSPI. It needs a supported Kerberos client other than SSPI.

Kerberos client	Default keytab file	GSS-API library file name	Notes
Windows MIT Kerberos client	`C:\WINDOWS\krb5kt`	`gssapi32.dll`or `gssapi64.dll`	The KRB5_KTNAME environment variable can be set before starting the database server to specify a different keytab file.
Windows CyberSafe Kerberos client	`C:\Program Files \CyberSafe\v5srvtab`	`gssapi32.dll`or `gssapi64.dll`	The CSFC5KTNAME environment variable can be set before starting the database server to specify a different keytab file.
Unix MIT Kerberos client	`/etc/krb5.keytab`	`libgssapi_krb5.so`[1]	The KRB5_KTNAME environment variable can be set before starting the database server to specify a different keytab file.

Kerberos client	Default keytab file	GSS-API library file name	Notes
Unix CyberSafe Kerberos client	`/krb5/v5srvtab`	`libgss.so`[1]	The CSFC5KTNAME environment variable can be set before starting the database server to specify a different keytab file.
Unix Heimdal Kerberos client	`/etc/krb5.keytab`	`libgssapi.so.1`[1]	

[1] These file names may vary depending on your operating system and Kerberos client version.

2.2.2 Setting up a Kerberos system to use with SAP IQ

Configure Kerberos authentication to be used with SAP IQ.

Prerequisites

You must be logged in to your computer using Kerberos authentication.

Context

Kerberos is a network authentication protocol that provides strong authentication and encryption using secret-key cryptography.

Procedure

1. If necessary, install and configure the Kerberos client software, including the GSS-API runtime library, on both the client and server.

   On Windows client computers using an Active Directory Key Distribution Center (KDC), SSPI can be used and you do not need to install the Kerberos client.
2. If necessary, create a Kerberos principal in the Kerberos KDC for each user.

   A Kerberos principal is a Kerberos user ID in the format `<user>`/`<instance>`@`<REALM>`, where /`<instance>` is optional. If you are already using Kerberos, the principal should already exist, so you do not need to create a Kerberos principal for each user.

   Principals are case sensitive and must be specified in the correct case. Mappings for multiple principals that differ only in case are not supported (for example, you cannot have mappings for both jjordan@MYREALM.COM and JJordan@MYREALM.COM).
3. Create a Kerberos principal in the KDC for the SAP IQ database server.

The default Kerberos principal for the database server has the format `<server-name>`@`<REALM>`, where `<server-name>` is the SAP IQ database server name. To use a different server principal, use the -kp server option. Principals are case significant, and `<server-name>` cannot contain multibyte characters, or the characters /, \, or @.

You must create a server service principal within the KDC because servers use a keytab file for KDC authentication. The keytab file is protected and encrypted.

4. Securely extract and copy the keytab for the principal `<server-name>`@`<REALM>` from the KDC to the computer running the SAP IQ database server. The default location of the keytab file depends on the Kerberos client and the platform. The keytab file's permissions should be set so that the SAP IQ server can read it, but unauthorized users do not have read permission.

Results

The Kerberos system is authenticated and configured to be used with SAP IQ.

Next Steps

Configure your SAP IQ database server and database to use Kerberos.

2.2.3 Configuring SAP IQ databases to use Kerberos

Configure databases to use Kerberos logins.

Prerequisites

You must have the SET ANY PUBLIC OPTION and MANAGE ANY USER system privileges.

You must already have Kerberos configured before SAP IQ can use it.

Context

The Kerberos login feature allows you to maintain a single user ID and password for database connections, operating systems, and network logins.

Procedure

1. Start the database server with the -krb or -kr option to enable Kerberos authentication, or use the -kl option to specify the location of the GSS-API library and enable Kerberos.
2. Change the public or temporary public option login_mode to a value that includes Kerberos. As database options apply only to the database in which they are found, different databases can have a different Kerberos login setting, even if they are loaded and running on the same database server. For example:

   ```
   SET OPTION PUBLIC.login_mode = 'Kerberos,Standard';
   ```

   > ⚠ Caution
   >
   > Setting the login_mode database option to Kerberos restricts connections to only those users who have been granted a Kerberos login mapping. Attempting to connect using a user ID and password generates an error unless you are a user with SYS_AUTH_DBA_ROLE compatibility role.

3. Create a database user ID for the client user. You can use an existing database user ID for the Kerberos login, as long as that user has the correct privileges. For example:

   ```
   CREATE USER "kerberos-user"
   IDENTIFIED BY abc123;
   ```

4. Execute a GRANT KERBEROS LOGIN TO statement to create a mapping from the client's Kerberos principal to an existing database user ID. For example:

   ```
   GRANT KERBEROS LOGIN TO "pchin@MYREALM.COM"
   AS USER "kerberos-user";
   ```

   To connect when a Kerberos principal is used that does not have a mapping, ensure the Guest database user ID exists and has a password.

5. Ensure the client user has already logged on (has a valid Kerberos ticket-granting ticket) using their Kerberos principal and that the client's Kerberos ticket has not expired. A Windows user logged in to a domain account already has a ticket-granting ticket, which allows them to authenticate to servers, providing their principal has enough permissions.

   A ticket-granting ticket is a Kerberos ticket encrypted with the user's password that is used by the Ticket Granting Service to verify the user's identity.

6. Connect from the client, specifying the KERBEROS connection parameter (Often KERBEROS=YES, but KERBEROS=SSPI or KERBEROS=`<GSS-API-library-file>` can also be used). If the user ID or password connection parameters are specified, they are ignored. For example:

   ```
   dbisql -c "KERBEROS=YES;Server=my_server_princ"
   ```

Results

The database is configured to use Kerberos authentication.

Example

A connection attempt using the following SQL statement is successful if the user logs in with a user profile name that matches a Kerberos login mapping for the default database on a database server:

```
CONNECT USING 'KERBEROS=YES';
```

The CONNECT statement can connect to a database if all the following conditions are true:

- A database server is currently running.
- The default database on the current database server is enabled to accept Kerberos authenticated connections.
- A Kerberos login mapping has been created for the user's current Kerberos principal.
- If the user is prompted with a window by the database server for more connection information (such as occurs when using Interactive SQL), the user clicks **OK** without providing more information.

Next Steps

You can use Kerberos authentication to connect from a client. Optionally, you can create a Kerberos login mapping.

2.2.4 Connections from an SAP Open Client or jConnect application

The database server accepts connections from an SAP Open Client or jConnect application.

- Set up Kerberos authentication.
- Configure SAP IQ to use Kerberos.
- Set up SAP Open Client or jConnect as you would for Kerberos authentication with Adaptive Server Enterprise. The server name must be the SAP IQ server's name and is case significant. You cannot connect using an alternate server name from Open Client or jConnect.

2.2.5 Connecting using SSPI for Kerberos logins on Windows

Connect using SSPI without a Kerberos client installed on the client computer.

Prerequisites

You must already have Kerberos configured before SAP IQ can use it. You must already have your database server and database configured to use Kerberos.

Context

In a Windows domain, SSPI can be used on Windows-based computers without a Kerberos client installed on the client computer. Windows domain accounts already have associated Kerberos principals.

SSPI can only be used by SAP IQ clients in the Kerberos connection parameter. SAP IQ database servers cannot use SSPI. They need a supported Kerberos client other than SSPI.

Procedure

Connect to the database from the client computer. For example:

```
dbisql -c "KERBEROS=SSPI;Server=my_server_princ"
```

When Kerberos=SSPI is specified in the connection string, a Kerberos login is attempted.

A connection attempt using the following SQL statement also succeeds, providing the user has logged on with a user profile name that matches a Kerberos login mapping for the default database on a database server:

```
CONNECT USING 'KERBEROS=SSPI';
```

Results

You can use SSPI for Kerberos authentication on Windows.

2.2.6 Troubleshooting: Kerberos connections

If you get unexpected errors when attempting to enable or use Kerberos authentication, it is recommended that you enable additional diagnostic messages on the database server and client.

Specifying the -z option when you start the database server, or using `CALL sa_server_option( 'DebuggingInformation', 'ON' )` if the server is already running includes additional diagnostic messages in the database server message log. The LogFile connection parameter writes client diagnostic messages to the specified file.

As an alternative to using the LogFile connection parameter, you can run the Ping utility (dbping) with the -z parameter. The -z parameter displays diagnostic messages that should help identify the cause of the connection problem.

Difficulties starting the database server

Symptom	Common solutions
"Unable to load Kerberos GSS-API library" message	● Ensure a Kerberos client is installed on the database server computer, including the GSS-API library. ● The database server -z output lists the name of the library that it is attempting to load. Verify the library name is correct. If necessary, use the -kl option to specify the correct library name. ● Ensure the directory and any supporting libraries is listed in the library path (%PATH% on Windows). ● If the database server -z output states the GSS-API library was missing entry points, then the library is not a supported Kerberos Version 5 GSS-API library.
"Unable to acquire Kerberos credentials for server name "`<server-name>`"" message	● Ensure there is a principal for `<server-name>`@`<REALM>` in the KDC. Principals are case sensitive, so ensure the database server name is in the same case as the user portion of the principal name. ● Ensure the name of the SAP IQ server is the primary/user portion of the principal. ● Ensure that the server's principal has been extracted to a keytab file and the keytab file is in the correct location for the Kerberos client. ● If the default realm for the Kerberos client on the database server computer is different from the realm in the server principal, use the -kr option to specify the realm in the server principal.
"Kerberos login failed" client error	● Check the database server diagnostic messages. Some problems with the keytab file used by the server are not detected until a client attempts to authenticate.

Troubleshooting Kerberos client connections

If the client got an error attempting to connect using Kerberos authentication:

Symptom	Common solutions
"Kerberos logins are not supported" error and the LogFile includes the message "Failed to load the Kerberos GSS-API library"	• Ensure a Kerberos client is installed on the client computer, including the GSS-API library. • The file specified by LogFile lists the name of the library that it is attempting to load. Verify that the library name is correct, and use the Kerberos connection parameter to specify the correct library name, if necessary. • Ensure that the directory including any supporting libraries is listed in the library path (%PATH% on Windows). • If the LogFile output states the GSS-API library was missing entry points, then the library is not a supported Kerberos Version 5 GSS-API library.
"Kerberos logins are not supported" error	• Ensure the database server has enabled Kerberos logins by specifying one or more of the -krb, -kl, or -kr server options. • Ensure Kerberos logins are supported on both the client and server platforms.
"Kerberos login failed" error	• Ensure the user is logged into Kerberos and has a valid ticket-granting ticket that has not expired. • Ensure the client computer and server computer both have their time synchronized to within less than 5 minutes.
"Login mode 'Kerberos' not permitted by login_mode setting" error	• The public or temporary public database option setting for the login_mode option must include the value Kerberos to allow Kerberos logins.
"The login ID '`<client-Kerberos-principal>`' has not been mapped to any database user ID"	• The Kerberos principal must be mapped to a database user ID using the GRANT KERBEROS LOGIN statement. Note the full client principal including the realm must be provided to the GRANT KERBEROS LOGIN statement, and principals which differ only in the instance or realm are treated as different. • Alternatively, if you want any valid Kerberos principal which has not be explicitly mapped to be able to connect, create the guest database user ID with a password using GRANT CONNECT.

2.2.7 Security: Use login modes to secure the database

There are several measures you can take to secure your database.

Setting the value of the login_mode option for a given database to allow a combination of Standard, Integrated, Kerberos, and LDAPUA logins using the SET OPTION statement permanently enables the specified types of logins for that database. When you enable Integrated, Kerberos, or LDAPUA logins for your database, you rely on the security model of the operating system or network. For example, the following statement permanently enables Standard and Integrated logins:

```
SET OPTION PUBLIC.login_mode = 'Standard,Integrated';
```

If the database is shut down and restarted, the option value remains the same and Integrated logins remain enabled.

Setting the login_mode option using SET TEMPORARY OPTION still allows user access via Integrated logins, but only until the database is shut down. The following statement changes the option value temporarily:

```
SET TEMPORARY OPTION PUBLIC.login_mode = 'Standard,Integrated';
```

If the permanent option value is Standard, the database will revert to that value when it is shut down.

Setting temporary public options can provide additional security for your database. If the database file is copied to another computer, then Integrated, Kerberos, and LDAPUA logins will not be enabled by default.

If a database contains sensitive information, the computer where the database files are stored should be protected from unauthorized access. Otherwise, the database files could be copied and unauthorized access to the data could be obtained on another computer.

To increase database security:

- Make passwords complex and difficult to guess.
- Strongly encrypt the database file using the AES encryption features of SAP IQ. The encryption key should be complex and difficult to guess.
- Set the permanent PUBLIC.login_mode database option to Standard. To enable Integrated or Kerberos logins, only the temporary public option should be changed each time the server is started. This ensures that only Standard logins are allowed if the database is copied.

2.3 Licensing Requirements for Kerberos

The Advanced Security Option (IQ_SECURITY) protects your environment against unauthorized access, and is required to use Kerberos authentication with SAP IQ.

2.4 PAM User Authentication

Pluggable Authentication Module (PAM) support allows you to write programs that rely on authentication independently from the underlying authentication scheme.

PAM authentication centralizes user authentication in a separate external system-wide module. If you already use PAM for other databases, such as Adaptive Server Enterprise, you can use the same credentials on all of your systems.

You can "plug in" any authentication mechanism into a PAM system by writing an authentication module. To configure PAM to use that module, add the module to the service name, the set of rules used in PAM authentication.

PAM User Authentication (PAMUA) is available on all supported UNIX and Linux platforms. To configure PAM for your platform, see your operating system documentation.

You must upgrade your database to a version that supports PAMUA, or create a new database, to use PAMUA.

In this section:

2.4.1 Enabling PAM User Authentication

Enable PAM user authentication in the database.

Prerequisites

- PAM is configured on the UNIX system. (See your operating system documentation.)

- Step 2 requires the SET ANY SECURITY OPTION system privilege.
- Steps 3 and 4 require the MANAGE ANY LOGIN POLICY system privilege.

Procedure

1. Configure PAM on your Linux or UNIX system. For this example, assume a PAM service name **PAM_Rule**.
2. Add the PAM value to the LOGIN_MODE database option:

   ```
   SET OPTION PUBLIC.login_mode = PAMUA
   ```
3. Create a login policy that assigns values to pam_service_name and (optionally) pam_failover_to_std:

   ```
   CREATE LOGIN POLICY pam_policy
   ```

   ```
   pam_service_name = PAM_Rule
   ```

   ```
   pam_failover_to_std = ON
   ```
4. Assign the login policy to the users of PAMUA:

   ```
   ALTER USER pam_userID LOGIN POLICY pam_policy
   ```

   You can also create new users and assign the login policy pam_policy to them with a `CREATE USER` statement.
5. Verify that users can log on using PAM user authentication. Use the `dbping` utility to test the database connection:

   ```
   dbping -c "uid=pam_userID;pwd=<pam_user_password>;
   links=tcpip(host=iq_server;port=6263);eng=IQdb" -d
   ```

2.4.2 Sample PAM Authorization Program

This sample program accepts credentials as command line parameters and uses PAM to authenticate them.

The program demonstrates basic authentication. It offers guidance for setting up and troubleshooting PAM authentication without the SAP IQ server.

This program was tested on RedHat Linux 6 using GCC 4.2; other UNIX platforms may require changes. Consult your system's PAM application programming interface for help compiling and running PAM client programs.

```
//*****************************
//Sample Application
//
//To compile this program, use:
//
// gcc pam_test.c
//
//*****************************
#include <security/pam_appl.h>
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int null_conv(int num_msg, const struct pam_message **msg, struct pam_response
**resp,
void *appdata_ptr)
{
   *resp=(struct pam_response*)appdata_ptr;
   return PAM_SUCCESS;
}
int authenticate(char *service, char *user, char *pass)
/**************************************************/
{
   pam_handle_t *pamh=NULL;
   struct pam_response *replay=(struct pam_response *)malloc( sizeoff(struct
pam_response) );
   struct pam_conv conv={nul_cov, (void*)reply };
   int retval=pam_start( service, user, &conv, &pamh );
   if( retval==PAM_SUCCESS){
        reply[0].resp=pass;
        reply[0].rep_retcode=0;
        retval=pam_authenticate( pamh, 0);
        pam_end( pamh, PAM_SUCCESS);
   }
   return (retval==PAM_SUCCESS?0:1);
}
int main(int argc, char *argv )
/*****************************/
{
   int retval;
   char *user, *pass, *service;
//
*****************************************************************************
//Accept command line parameters for username, password, and optional
servicename.
//
*****************************************************************************
   if( argc<3||argc >4){
        fprintf(stderr, "Usage: login <username> <password> [<servicename> ]\n");
        exit(1);
   }
   user=argv[1];
   pass=strdup( argv[2]);
   service=(argc>=4) ? argv[3]:"system-auth";
   retval=authenticate( service, user, pass );
   if (retval==PAM_SUCCESS){
         fprintf(stdout, "Authenticated\n");)
   } else {
         fprintf(stdout, "Not Authenticated\n");
   }
   return retval;
}
```

2.4.3 Sample PAM Configuration

Configure PAM to delegate user authentication to the system authentication module, which is also used by other operation system services like rlogin, chsh, and gdm.

This example reuses the common rlogin PAM module to allow for system level authentication. The UNIX user ID used for the rlogin tool can also be used by the database server for UNIX account authentication. SAP recommends that you use a copy of the rlogin.

1. Verify that PAM is installed correctly on the UNIX machine using the `pamclient` command, if that utility is available. If `pamclient` is not installed, proceed with the next step.
2. Connect to the database as owner, and enter:

   ```
   SET TEMPORARY option PUBLIC.LOGIN_MODE = 'PAMUA,STANDARD'
   CREATE LOGIN POLICY usepam
   PAM_SERVICE_NAME = rlogin
   PAM_FAILOVER_TO_STD = ON
   ```

   PAM_SERVICE_NAME is set to the name of the configured PAM service. Please refer to the PAM configuration guide to set up and configure a PAM service. On Linux, PAM service configuration files are in `/etc/pam.d`.
3. Create a user with the login policy:

   ```
   CREATE USER Company-uid LOGIN POLICY usepam
   ```

4. Log in using the `Company-uid` user ID and password.
5. When finished, drop the user and login policy:

   ```
   DROP USER Company-uid
   DROP LOGIN policy usepam
   ```

3 Advanced Security Options in SAP IQ

The SAP IQ Advanced Security Option supports column encryption, and LDAP and Kerberos authentication for database connections, operating system logins, and network logins. The Advanced Security Option is a separately licensed SAP IQ option.

In this section:

Column Encryption in SAP IQ [page 241]
SAP IQ supports user-encrypted columns.

Kerberos Authentication Support in SAP IQ [page 279]
SAP IQ supports Kerberos authentication, a login feature that allows you to maintain a single user ID and password for both database connections and operating system and network logins.

LDAP User Authentication Support in SAP IQ [page 279]
You can integrate SAP IQ into any existing enterprise-wide directory access framework based on Lightweight Directory Access Protocol (LDAP), a widely accepted international standard.

3.1 Column Encryption in SAP IQ

SAP IQ supports user-encrypted columns.

Strong encryption of the SAP IQ database file uses a 128-bit algorithm and a security key. The data is unreadable and virtually undecipherable without the key. The algorithm supported is described in FIPS-197, the Federal Information Processing Standard for the Advanced Encryption Standard.

SAP IQ supports user-encrypted columns with the `AES_ENCRYPT` and `AES_DECRYPT` functions and the `LOAD TABLE ENCRYPTED` clause. These functions permit explicit encryption and decryption of column data via calls from the application. Encryption and decryption key management is the responsibility of the application.

Certain database options affect column encryption.

In this section:

Licensing Requirements for Column Encryption [page 242]
The Advanced Security Option (IQ_SECURITY) is required to use user-encrypted columns with SAP IQ.

Definitions of Encryption Terms [page 242]
Definitions of terms used when describing encryption of stored data.

Data Types for Encrypted Columns [page 243]
The data types supported for encrypted columns and working with these data types.

AES_ENCRYPT Function [String] [page 245]
Encrypts the specified values using the supplied encryption key, and returns a `VARBINARY` or `LONG VARBINARY`.

AES_DECRYPT Function [String] [page 249]

Decrypts the string using the supplied key, and returns, by default, a `VARBINARY` or `LONG BINARY`, or the original plaintext type.

LOAD TABLE ENCRYPTED Clause [page 250]

The `LOAD TABLE` statement supports the column-spec keyword `ENCRYPTED`.

String Comparisons on Encrypted Text [page 270]

If data is case-insensitive, or uses a collation other than ISO_BINENG, you must decrypt ciphertext columns to perform string comparisons.

Database Options for Column Encryption [page 271]

Certain SAP IQ database option settings affect column encryption and decryption; the default settings are not optimal for most column encryption operations.

Encryption and Decryption Example [page 273]

An example using the `AES_ENCRYPT` and `AES_DECRYPT` functions, written in commented SQL.

Related Information

Database Options for Column Encryption [page 271]

3.1.1 Licensing Requirements for Column Encryption

The Advanced Security Option (IQ_SECURITY) is required to use user-encrypted columns with SAP IQ.

3.1.2 Definitions of Encryption Terms

Definitions of terms used when describing encryption of stored data.

- plaintext – data in its original, intelligible form. Plaintext is not limited to string data, but is used to describe any data in its original representation.
- ciphertext – data in an unintelligible form that preserves the information content of the plaintext form.
- encryption – a reversible transformation of data from plaintext to ciphertext. Also known as enciphering.
- decryption – the reverse transformation of ciphertext back to plaintext. Also known as deciphering.
- key – a number used to encrypt or decrypt data. Symmetric-key encryption systems use the same key for both encryption and decryption. Asymmetric-key systems use one key for encryption and a different (but mathematically related) key for decryption. The SAP IQ interfaces accept character strings as keys.
- Rijndael – pronounced “reign dahl.” A specific encryption algorithm that supports a variety of key and block sizes. The algorithm was designed to use simple whole-byte operations and thus is relatively easy to implement in software.
- AES – the Advanced Encryption Standard, a FIPS-approved cryptographic algorithm for the protection of sensitive (but unclassified) electronic data. AES adopted the Rijndael algorithm with restrictions on the block sizes and key lengths. AES is the algorithm supported by SAP IQ.

3.1.3 Data Types for Encrypted Columns

The data types supported for encrypted columns and working with these data types.

In this section:

3.1.3.1 Supported Data Types

The first parameter of the `AES_ENCRYPT` function must be one of the supported data types.

CHAR	NUMERIC
VARCHAR	FLOAT
TINYINT	REAL
SMALLINT	DOUBLE
INTEGER	DECIMAL
BIGINT	DATE
BIT	TIME
BINARY	DATETIME
VARBINARY	TIMESTAMP
UNSIGNED INT	SMALLDATETIME
UNSIGNED BIGINT	

The `LOB` data type is not currently supported for SAP IQ column encryption.

3.1.3.2 Preservation of Data Types

SAP IQ ensures that the original data type of plaintext is preserved when decrypting data, if the AES_DECRYPT function is given the data type as a parameter, or is within a CAST function.

SAP IQ compares the target data type of the CAST function with the data type of the originally encrypted data. If the two data types do not match, you see a -1001064 error that includes details about the original and target data types.

For example, given an encrypted VARCHAR(1) value and this valid decryption statement:

```
SELECT AES_DECRYPT ( thecolumn, ‘theKey’,
VARCHAR(1) ) FROM thetable
```

If you attempt to decrypt the data using:

```
SELECT AES_DECRYPT ( thecolumn, ‘theKey’,
SMALLINT ) FROM thetable
```

the error returned is:

```
Decryption error: Incorrect CAST type smallint(5,0)
for decrypt data of type varchar(1,0).
```

This data type check is made only when the CAST or the data type parameter are supplied. Otherwise, the query returns the ciphertext as binary data.

When using the AES_ENCRYPT function on literal constants, as in this statement:

```
INSERT INTO t (cipherCol) VALUES (AES_ENCRYPT (1, ‘key’))
```

the data type of 1 is ambiguous; it can be a TINYINT, SMALLINT, INTEGER, UNSIGNED INT, BIGINT, UNSIGNED BIGINT, or possibly other data types.

You should explicitly use the CAST function to resolve any potential ambiguity, as in:

```
INSERT INTO t (cipherCol)
VALUES ( AES_ENCRYPT (CAST (1 AS UNSIGNED INTEGER), ‘key’))
```

Explicitly converting the data type using the CAST function when encrypting data prevents problems using the CAST function when the data is decrypted.

There is no ambiguity if the data being encrypted is from a column, or if the encrypted data was inserted by LOAD TABLE.

3.1.3.3 Effect of Different Data Types on Ciphertext

To produce identical ciphertext for different datatypes, cast the input of AES_ENCRYPT to the same data type to produce identical ciphertext.

The ciphertext produced by AES_ENCRYPT differs for two different data types given the same input value and same key. A join of two ciphertext columns that holds encrypted values of two different data types may therefore not return identical results.

For example, assume:

```
CREATE TABLE tablea(c1 int, c2 smallint);
INSERT INTO tablea VALUES (100,100);
```

The value AES_ENCRYPT(c1, 'key') differs from AES_ENCRYPT(c2,'key') and the value AES_ENCRYPT(c1,'key') differs from AES_ENCRYPT(100,'key').

To resolve this issue, cast the input of AES_ENCRYPT to the same data type. For example, the results of these code fragments are the same:

```
AES_ENCRYPT(c1, 'key');
```

```
AES_ENCRYPT(CAST(c2 AS INT), 'key');
```

```
AES_ENCRYPT(CAST(100 AS INT), 'key');
```

Related Information

AES_ENCRYPT Function [String] [page 245]

3.1.4 AES_ENCRYPT Function [String]

Encrypts the specified values using the supplied encryption key, and returns a VARBINARY or LONG VARBINARY.

Syntax

```
AES_ENCRYPT( <string-expression>, <key> )
```

Parameters

`<string-expression>` – the data to be encrypted. You can also pass binary values to `AES_ENCRYPT`. This parameter is case-sensitive, even in case-insensitive databases.

`<key>` – the encryption key used to encrypt the `<string-expression>`. To obtain the original value, also use the same key to decrypt the value. This parameter is case-sensitive, even in case-insensitive databases.

As you should for most passwords, choose a key value that is difficult to guess. Choose a value that is at least 16 characters long, contains a mix of uppercase and lowercase letters, and includes numbers and special characters. You need this key each time you want to decrypt the data.

> ⚠ Caution
>
> Protect your key; store a copy of your key in a safe location. If you lose your key, encrypted data becomes completely inaccessible and unrecoverable.

Usage

`AES_ENCRYPT` returns a `VARBINARY` value, which is at most 31 bytes longer than the input `<string-expression>`. The value returned by this function is the ciphertext, which is not human-readable. You can use the `AES_DECRYPT` function to decrypt a `<string-expression>` that was encrypted with the `AES_ENCRYPT` function. To successfully decrypt a `<string-expression>`, use the same encryption key and algorithm used to encrypt the data. If you specify an incorrect encryption key, an error is generated.

If you are storing encrypted values in a table, the column should be of data type `VARBINARY` or `VARCHAR`, and greater than or equal to 32 bytes, so that character set conversion is not performed on the data. (Character set conversion prevents data decryption.) If the length of the `VARBINARY` or `VARCHAR` column is fewer than 32 bytes, the `AES_DECRYPT` function returns an error.

The result data type of an `AES_ENCRYPT` function may be a `LONG BINARY`. If you use `AES_ENCRYPT` in a `SELECT INTO` statement, you must have an Unstructured Data Analytics Option license, or use `CAST` and set `AES_ENCRYPT` to the correct data type and size.

Standards and Compatibility

- SQL – vendor extension to ISO/ANSI SQL grammar.
- Sybase – not supported by SAP ASE.

In this section:

REPLACE Function [String] [page 247]
Replaces all occurrences of a substring with another substring.

Related Information

AES_DECRYPT Function [String] [page 249]
Encryption and Decryption Example [page 273]
LOAD TABLE ENCRYPTED Clause [page 250]
Effect of Different Data Types on Ciphertext [page 245]
Data Types for Encrypted Columns [page 243]

3.1.4.1 REPLACE Function [String]

Replaces all occurrences of a substring with another substring.

Syntax

```
REPLACE ( <original-string>, <search-string>, <replace-string >)
```

Parameters

If any argument is NULL, the function returns NULL.

Parameter	Description
original-string	The string to be searched. This string can be any length.
search-string	The string to be searched for and replaced with `<replace-string>`. This string is limited to 255 bytes. If `<search-string>` is an empty string, the original string is returned unchanged.
replace-string	The replacement string, which replaces `<search-string>`. This can be any length. If `<replace-string>` is an empty string, all occurrences of `<search-string>` are deleted.

Returns

LONG VARCHAR

LONG NVARCHAR

i Note

The result data type is a LONG VARCHAR. If you use REPLACE in a SELECT INTO statement, you must have an Unstructured Data Analytics Option license or use CAST and set REPLACE to the correct data type and size.

Remarks

The result data type of a REPLACE function is a LONG VARCHAR. If you use REPLACE in a SELECT INTO statement, you must have an Unstructured Data Analytics Option license, or use CAST and set REPLACE to the correct data type and size.

There are two ways to work around this issue:

- Declare a local temporary table, then perform an INSERT:

  ```
  DECLARE local temporary table #mytable
    (name_column char(10)) on commit preserve rows;
  INSERT INTO #mytable SELECT REPLACE(name,'0','1')   FROM dummy_table01;
  ```

- Use CAST:

  ```
  SELECT CAST(replace(name, '0', '1') AS Char(10)) into #mytable from
  dummy_table01;
  ```

If you need to control the width of the resulting column when< replace-string> is wider than <search-string>, use the CAST function. For example:

```
CREATE TABLE aa(a CHAR(5));
INSERT INTO aa VALUES(‘CCCCC’);
COMMIT;
SELECT a, CAST(REPLACE(a,’C’,’ZZ’) AS CHAR(5)) FROM aa;
```

Standards and Compatibility

- SQL—Vendor extension to ISO/ANSI SQL grammar.
- SAP Database Products—Compatible with SAP ASE.

Example

The following statement returns the value “xx.def.xx.ghi:”

```
SELECT REPLACE( 'abc.def.abc.ghi', 'abc', 'xx' ) FROM iq_dummy
```

The following statement generates a result set containing `ALTER PROCEDURE` statements which, when executed, repair stored procedures that reference a table that has been renamed. (To be useful, the table name must be unique.)

```
SELECT REPLACE(
  replace(proc_defn,'OldTableName','NewTableName'),
  'create procedure',
  'alter procedure')
FROM SYS.SYSPROCEDURE
WHERE proc_defn LIKE '%OldTableName%'
```

Use a separator other than the comma for the `LIST` function:

```
SELECT REPLACE( list( table_id ), ',', '--')
FROM  SYS.ISYSTAB
WHERE table_id <= 5
```

3.1.5 AES_DECRYPT Function [String]

Decrypts the string using the supplied key, and returns, by default, a `VARBINARY` or `LONG BINARY`, or the original plaintext type.

Syntax

```
AES_DECRYPT( <string-expression>, <key> [, <data-type> ] )
```

Parameters

`<string-expression>` – the string to be decrypted. You can also pass binary values to this function. This parameter is case sensitive, even in case-insensitive databases.

`<key>` – the encryption key required to decrypt the `<string-expression>`. To obtain the original value that was encrypted, the key must be the same encryption key that was used to encrypt the `<string-expression>`. This parameter is case-sensitive, even in case-insensitive databases.

> ⚠ Caution
>
> Protect your key; store a copy of your key in a safe location. If you lose your key, the encrypted data becomes completely inaccessible and unrecoverable.

`<data-type>` – this optional parameter specifies the data type of the decrypted `<string-expression>` and must be the same data type as the original plaintext.

If you do not use a `CAST` statement while inserting data using the `AES_ENCRYPT` function, you can view the same data using the `AES_DECRYPT` function by passing `VARCHAR` as the `<data-type>`. If you do not pass `<data-type>` to `AES_DECRYPT`, `VARBINARY` data type is returned.

Usage

You can use the AES_DECRYPT function to decrypt a <string-expression> that was encrypted with the AES_ENCRYPT function. This function returns a VARBINARY or LONG VARBINARY value with the same number of bytes as the input string, if no data type is specified. Otherwise, the specified data type is returned.

To successfully decrypt a <string-expression>, you must use the same encryption key that was used to encrypt the data. An incorrect encryption key returns an error.

Example

Decrypt the password of a user from the user_info table.

```
SELECT AES_DECRYPT(user_pwd, '8U3dkA', CHAR(100))
FROM user_info;
```

Standards and Compatibility

- SQL – vendor extension to ISO/ANSI SQL grammar.
- Sybase – not supported by SAP ASE.

Related Information

AES_ENCRYPT Function [String] [page 245]
Encryption and Decryption Example [page 273]
LOAD TABLE ENCRYPTED Clause [page 250]
Data Types for Encrypted Columns [page 243]

3.1.6 LOAD TABLE ENCRYPTED Clause

The LOAD TABLE statement supports the column-spec keyword ENCRYPTED.

The <column-specs> must follow the column name in a LOAD TABLE statement in this order:

- <format-specs>
- <null-specs>
- <encrypted-specs>

Syntax

```
| ENCRYPTED(<data-type> ‘<key-string>’ [, ‘<algorithm-string>’ ] )
```

Parameters

data-type the data type that the input file field should be converted to as input to the `AES_ENCRYPT` function. `<data-type>` should be the same as the data type of the output of the `AES_DECRYPT` function.

key-string the encryption key used to encrypt the data. This key must be a string literal. To obtain the original value, use the same key to decrypt the value. This parameter is case-sensitive, even in case-insensitive databases.

As you should for most passwords, choose a key value that cannot be easily guessed. Choose a value for that is at least 16 characters long, contains a mix of uppercase and lowercase letters, and includes numbers and special characters. You will need this key each time you want to decrypt the data.

> ⚠ Caution
>
> Protect your key; store a copy of your key in a safe location. A lost key results in the encrypted data becoming completely inaccessible, from which there is no recovery.

algorithm-string the algorithm used to encrypt the data. This parameter is optional, but data must be encrypted and decrypted using the same algorithm. Currently, AES is the default, as it is the only supported algorithm. AES is a block encryption algorithm chosen as the new Advanced Encryption Standard (AES) for block ciphers by the National Institute of Standards and Technology (NIST).

Usage

The `ENCRYPTED` column specification allows you to specify the encryption key and, optionally, the algorithm to use to encrypt the data that is loaded into the column. The target column for this load should be `VARBINARY`. Specifying other data types returns an error.

Example

```
LOAD TABLE <table_name>
(
<plaintext_column_name>,
<a_ciphertext_column_name>
NULL('nil')
ENCRYPTED(varchar(6),'tHefiRstkEy') ,
<another_encrypted_column>
ENCRYPTED(bigint,'thEseconDkeY','AES')
)
```

```
FROM '/path/to/the/input/file'
FORMAT ascii
DELIMITED BY ';'
ROW DELIMITED BY '\0xa'
QUOTES OFF
ESCAPES OFF
```

where the format of the input file for the LOAD TABLE statement is:

```
a;b;c;
d;e;f;
g;h;i;
```

In this section:

LOAD TABLE Statement [page 252]
Imports data into a database table from an external file.

Related Information

AES_ENCRYPT Function [String] [page 245]
AES_DECRYPT Function [String] [page 249]
Encryption and Decryption Example [page 273]
Data Types for Encrypted Columns [page 243]

3.1.6.1 LOAD TABLE Statement

Imports data into a database table from an external file.

Syntax

```
LOAD [ INTO ] TABLE [ <owner>.]<table-name>
   … ( load-specification [page 253] [, …] )
   … { FROM | USING [ CLIENT ] FILE }
        { '<filename>-<string>' | <filename>-<variable> } [, …]
   … [ CHECK CONSTRAINTS { ON | OFF } ]
   … [ DEFAULTS { ON | OFF } ]
   …   QUOTES { ON | OFF } ]
   … [ QUOTE <enclosure_character> ]
   … [ QUOTE ESCAPE '<escape_character>' ]
   … ESCAPES OFF
   … [ FORMAT { ascii | binary | bcp  csv } ]
   … [ DELIMITED BY '<string>' ]
   … [ STRIP { OFF | RTRIM } ]
   … [ WITH CHECKPOINT { ON | OFF } ]
   … [ BYTE ORDER { NATIVE | HIGH | LOW } ]
   … [ LIMIT <number-of-rows> ]
   … [ NOTIFY <number-of-rows> ]
   … [ ON FILE ERROR { ROLLBACK | FINISH | CONTINUE } ]
   … [ PREVIEW { ON | OFF } ]
   … [ ROW DELIMITED BY '<delimiter-string>' ]
   … [ SKIP <number-of-rows> ]
   … [ HEADER SKIP [ ALL ] <number> [ HEADER DELIMITED BY '<string>' ] ]
   … [ WORD SKIP <number> ]
   … [ ON PARTIAL INPUT ROW { ROLLBACK | CONTINUE } ]
   … [ IGNORE CONSTRAINT constraint-type [page 253] [, …] ]
   … [ MESSAGE LOG '<string>' ROW LOG '<string>' [ ONLY LOG log-what [page 253]
[, …] ]
   … [ LOG DELIMITED BY '<string>' ]
load-specification
   { <column-name> [ column-spec [page 253] ]
      | FILLER ( filler-type [page 253] ) }
column-spec
   { ASCII ( <input-width> )
   | BINARY [ WITH NULL BYTE ]
   | PREFIX { 1 | 2 | 4 }
   | '<delimiter-string>'
   | DATE ( <input-date-format> )
   | DATETIME ( <input-datetime-format> )
   | FILE NAME
   | ENCRYPTED ( <data-type> '<key-string>' [, '<algorithm-string>' ] )
   | DEFAULT <default-value> }
   [ NULL ( { BLANKS | ZEROS | '<literal>',  …} )
filler-type
   { <input-width>
   | PREFIX { 1 | 2 | 4 }
   | '<delimiter-string>'
   }
constraint-type
   { CHECK <integer>
   | UNIQUE <integer>
   | NULL <integer>
   | <FOREIGN KEY> <integer>
   | DATA VALUE <integer>
   | ALL <integer>
   }
log-what
   { CHECK
   | ALL
   | NULL
   | UNIQUE
   | DATA VALUE
   | FOREIGN KEY
   | WORD
   }
```

Parameters

- **FROM** identifies one or more files from which to load data. To specify more than one file, use a comma to separate each `filename-string`. The `<filename-string>` is passed to the server as a string. The string is therefore subject to the same formatting requirements as other SQL strings. To indicate directory paths on Windows, the backslash character (\) must be represented by two backslashes (\\). Therefore, the statement to load data from the file `c:\temp\input.dat` into the `Employees` table is:

  ```
  LOAD TABLE Employees
  FROM 'c:\\temp\\input.dat' …
  ```

  The path name is relative to the database server, not to the client application. If you are running the statement on a database server on some other computer, the directory names refers to directories on the server machine, not on the client machine. When loading a multiplex database, use absolute (fully qualified) paths in all file names. Do not use relative path names.

Because of resource constraints, SAP IQ does not guarantee that all the data can be loaded. If resource allocation fails, the entire load transaction is rolled back. The files are read one at a time, and processed in the order specified in the FROM clause. Any SKIP or LIMIT value only applies in the beginning of the load, not for each file.

The LOAD TABLE FROM clause is deprecated, but may be used to specify a file that exists on the server. This example loads data from the file a.inp on a client computer:

```
LOAD TABLE t1(c1,c2,filler(30))
USING CLIENT FILE 'c:\\client-data\\a.inp'
QUOTES OFF ESCAPES OFF
IGNORE CONSTRAINT UNIQUE 0, NULL 0
MESSAGE LOG 'c:\\client-data\\m.log'
ROW LOG 'c:\\client-data\\r.log'
ONLY LOG UNIQUE
```

- **USING** USING FILE loads one or more files from the server. This clause is synonymous with specifying the FROM `<filename>` clause. USING CLIENT FILE bulk loads one or more files from a client. The character set of the file on the client side must be the same as the server collation. SAP IQ serially processes files in the file list. Each file is locked in read mode as it is processed, then unlocked. Client-side bulk loading incurs no administrative overhead, such as extra disk space, memory or network-monitoring daemon requirements, but does forces single threaded processing for each file. When bulk loading large objects, the USING CLIENT FILE clause applies to both primary and secondary files.

  The `LOAD TABLE` statement can load compressed client and server files in the gzip format only. Any file with an extension ".gz" or ".gzip" is assumed to be a compressed file. Named pipes or secondary files are not supported during a compressed file load. Compressed files and uncompressed files can be specified in the same `LOAD TABLE` statement. Each compressed file in a load is processed by one thread.

  During client-side loads, the IGNORE CONSTRAINT log files are created on the client host and any error while creating the log files causes the operation to roll back.

  Client-side bulk loading is supported by Interactive SQL and ODBC/JDBC clients using the Command Sequence protocol. It is not supported by clients using the TDS protocol. For data security over a network, use Transport Layer Security. To control who can use client-side bulk loads, use the secure feature (`-sf`) server startup switch, enable the `ALLOW_READ_CLIENT_FILE` database option, and the READ CLIENT FILE access control.
- **CHECK CONSTRAINTS** evaluates check constraints, which you can ignore or log. CHECK CONSTRAINTS defaults to ON. Setting CHECK CONSTRAINTS OFF causes SAP IQ to ignore all check constraint violations. This can be useful, for example, during database rebuilding. If a table has check constraints that call user-defined functions that are not yet created, the rebuild fails unless this option is set to OFF.

  This option is mutually exclusive to the following options. If any of these options are specified in the same load, an error results:
  - IGNORE CONSTRAINT ALL
  - IGNORE CONSTRAINT CHECK
  - LOG ALL
  - LOG CHECK
- **DEFAULTS** uses a column's default value. This option is ON by default. If the DEFAULTS option is OFF, any column not present in the column list is assigned NULL. The setting for the DEFAULTS option applies to all column DEFAULT values, including AUTOINCREMENT.

- **QUOTE** for TEXT data only; identifies the enclosure character to be placed around string values. If not specified, the default QUOTE character is either a single (') or double (") quotation mark, depending on what is used in the field. If QUOTES OFF is defined, QUOTE is ignored. If the specified `<enclosure_character>` is multibyte, only the first byte is used; the remaining bytes are ignored.
- **QUOTE ESCAPE** specifies the escape character used in the data. If not specified, the default QUOTE ESCAPE character is the value of QUOTE. For example, if QUOTE is defined as percent (%), but QUOTE ESCAPE is not defined, the default value for QUOTE ESCAPE becomes %. If neither QUOTE ESCAPE nor QUOTE are defined, QUOTE defaults to either a single (') or double (") quotation mark, depending on what is used in the field, and QUOTE ESCAPE defaults to match QUOTE. If the specified ESCAPE character is multibyte, only the first byte is used; the remaining bytes are ignored.

  If QUOTES ON and QUOTE ESCAPE is not defined, single quote becomes the ESCAPE character and must be escaped by another quote.

  > Note
  >
  > QUOTE ESCAPE clause not supported when loading CSV file into IQ catalog store tables. `Syntax not supported` message appears.

- **QUOTES** indicates that input strings are enclosed in quote characters. QUOTES is an optional parameter and is ON by default. The first such character encountered in a string is treated as the quote character for the string. String data must be terminated with a matching quote. With QUOTES ON, column or row delimiter characters can be included in the column value. Leading and ending quote characters are assumed not to be part of the value and are excluded from the loaded data value.

  To include a quote character in a value with QUOTES ON, use two quotes. For example, this line includes a value in the third column that is a single quote character:

  ```
  '123 High Street, Anytown', '(715)398-2354',''''
  ```

  With STRIP turned on (the default), trailing blanks are stripped from values before they are inserted. Trailing blanks are stripped only for non-quoted strings. Quoted strings retain their trailing blanks. Leading blank or TAB characters are trimmed only when the setting is ON.

  The data extraction facility provides options for handling quotes (`TEMP_EXTRACT_QUOTES`, `TEMP_EXTRACT_QUOTES_ALL`, and `TEMP_EXTRACT_QUOTE`). If you plan to extract data to be loaded into an IQ main store table and the string fields contain column or row delimiter under default ASCII extraction, use the `TEMP_EXTRACT_BINARY` option for the extract and the FORMAT binary and QUOTES OFF options for `LOAD TABLE`.

  Limits:

  - QUOTES ON applies only to column-delimited ASCII fields.
  - With QUOTES ON, the first character of a column delimiter or row terminator cannot be a single or double quote mark.
  - QUOTES ON forces single threaded processing for a given file.
  - The QUOTES option does not apply to loading binary large object (BLOB) or character large object (CLOB) data from the secondary file, regardless of its setting. A leading or trailing quote is loaded as part of CLOB data. Two consecutive quotes between enclosing quotes are loaded as two consecutive quotes with the QUOTES ON option.
  - SAP ASE BCP does not support the QUOTES option. All field data is copied in or out equivalent to the QUOTES OFF setting. As QUOTES ON is the default setting for the SAP IQ `LOAD TABLE` statement, you must specify QUOTES OFF when importing ASE data from BCP output to an SAP IQ table.

Exceptions:

  - If LOAD TABLE encounters any nonwhite characters after the ending quote character for an enclosed field, this error is reported and the load operation is rolled back:

    ```
    Non-SPACE text found after ending quote character for
    an enclosed field.
    SQLSTATE: QTA14    SQLCODE: -1005014L
    ```
  - With QUOTES ON, if a single or double quote is specified as the first character of the column delimiter, an error is reported and the load operation fails:

    ```
    Single or double quote mark cannot be the 1st character
    of column delimiter or row terminator with QUOTES option
    ON.
    SQLSTATE: QCA90    SQLCODE: -1013090L
    ```
- **ESCAPES** if you omit a <column-spec> definition for an input field and ESCAPES is ON (the default), characters following the backslash character are recognized and interpreted as special characters by the database server. You can include newline characters as the combination \n, and other characters as hexadecimal ASCII codes, such as \x09 for the tab character. A sequence of two backslash characters (\ \ ) is interpreted as a single backslash. For SAP IQ, you must set ESCAPES OFF.
- **FORMAT** SAP IQ supports ASCII and binary input fields. The format is usually defined by the <column-spec> described above. If you omit that definition for a column, by default SAP IQ uses the format defined by this option. Input lines are assumed to have ASCII (the default) or binary fields, one row per line, with values separated by the column delimiter character.

  A row in a CSV file must be terminated either by a row deliminator (default newline) or a column deliminator (default coma) followed by a row delimiter. The maximum size of a delimiter is 4 bytes. An error message appears if the deliminator exceeds 4 bytes.

  A CSV file may contain partial rows, defined as any row with the number of fields less than the number of columns specified (either explicitly or implicitly) in the LOAD TABLE statement. All fields missing from a partial rows are assigned a NULL value. If the column is not nullable, an error message appears, and no data is imported.

  If a table has K columns, a CSV file has M fields, and the LOAD TABLE statement indicates N columns (either explicitly or implicitly), when N <= K and N < M, columns missing from the column list in the load statement are assigned default values.

  SAP IQ also accepts data from BCP character files as input to the LOAD TABLE command.

  - The BCP data file loaded into SAP IQ tables using the LOAD TABLE FORMAT BCP statement must be exported (BCP OUT) in cross-platform file format using the -c option.
  - For FORMAT BCP, the default column delimiter for the LOAD TABLE statement is <tab> and the default row terminator is <newline>.
  - For FORMAT BCP, the last column in a row must be terminated by the row terminator, not by the column delimiter. If the column delimiter is present before the row terminator, then the column delimiter is treated as a part of the data.
  - Data for columns that are not the last column in the load specification must be delimited by the column delimiter only. If a row terminator is encountered before a column delimiter for a column that is not the last column, then the row terminator is treated as a part of the column data.
  - Column delimiter can be specified via the DELIMITED BY clause. For FORMAT BCP, the delimiter must be less than or equal to 10 characters in length. An error is returned, if the delimiter length is more than 10.

- For FORMAT BCP, the load specification may contain only column names, NULL, and ENCRYPTED. An error is returned, if any other option is specified in the load specification.
  For example, these `LOAD TABLE` load specifications are valid:

  ```
  LOAD TABLE x( c1, c2 null(blanks), c3 )
  FROM 'bcp_file.bcp'
  FORMAT BCP
  …
  ```

  ```
  LOAD TABLE x( c1 encrypted(bigint,'KEY-ONE','aes'), c2, c3 )
  FROM 'bcp_file.bcp'
  FORMAT BCP
  …
  ```

- **DELIMITED BY** if you omit a column delimiter in the `<column-spec>` definition, the default column delimiter character is a comma. You can specify an alternative column delimiter by providing a single ASCII character or the hexadecimal character representation. The DELIMITED BY clause is:

  ```
  … DELIMITED BY '\x09' …
  ```

  To use the newline character as a delimiter, you can specify either the special combination '\n' or its ASCII value '\x0a'. Although you can specify up to four characters in the column-spec `<delimiter-string>`, you can specify only a single character in the DELIMITED BY clause.
- **STRIP** determines whether unquoted values should have trailing blanks stripped off before they are inserted. The `LOAD TABLE` command accepts these STRIP keywords:
  - `STRIP OFF` – do not strip off trailing blanks.
  - `STRIP RTRIM` – strip trailing blanks.
  - `STRIP ON` – deprecated. Use STRIP RTRIM.

  With STRIP turned on (the default), SAP IQ strips trailing blanks from values before inserting them. This is effective only for VARCHAR data. STRIP OFF preserves trailing blanks.

  Trailing blanks are stripped only for unquoted strings. Quoted strings retain their trailing blanks. If you do not require blank sensitivity, you can use the FILLER option as an alternative to be more specific in the number of bytes to strip, instead of all the trailing spaces. STRIP OFF is more efficient for SAP IQ, and it adheres to the ANSI standard when dealing with trailing blanks. (CHAR data is always padded, so the STRIP option only affects VARCHAR data.)

  The STRIP option applies only to variable-length non-binary data and does not apply to ASCII fixed-width inserts. For example, assume this schema:

  ```
  CREATE TABLE t( c1 VARCHAR(3) );
  LOAD TABLE t( c1 ',' ) …….. STRIP RTRIM    // trailing blanks trimmed
  LOAD TABLE t( c1 ',' ) …….. STRIP OFF      // trailing blanks not trimmed
  LOAD TABLE t( c1 ASCII(3) ) … STRIP RTRIM    // trailing blanks not trimmed
  LOAD TABLE t( c1 ASCII(3) ) … STRIP OFF      // trailing blanks trimmed
  LOAD TABLE t( c1 BINARY ) ….. STRIP RTRIM    // trailing blanks trimmed
  LOAD TABLE t( c1 BINARY ) ….. STRIP OFF      // trailing blanks trimmed
  ```

  Trailing blanks are always trimmed from binary data.
- **WITH CHECKPOINT** determines whether SAP IQ performs a checkpoint. This option is useful only when loading SAP SQL Anywhere tables in an SAP IQ database. The default setting is OFF. If this clause is set to ON, a checkpoint is issued after successfully completing and logging the statement. If the server fails after a connection commits and before the next checkpoint, the data file used to load the table must be present

for the recovery to complete successfully. However, if WITH CHECKPOINT ON is specified, and recovery is subsequently required, the data file need not be present at the time of recovery.

The data files are required, regardless of what is specified for this clause, if the database becomes corrupt and you need to use a backup and apply the current log file.

> ⚠ Caution
>
> If you set the database option CONVERSION_ERROR to OFF, you may load bad data into your table without any error being reported. If you do not specify WITH CHECKPOINT ON, and the database needs to be recovered, the recovery may fail as CONVERSION_ERROR is ON (the default value) during recovery. It is recommended that you do not load tables when CONVERSION_ERROR is set to OFF and WITH CHECKPOINT ON is not specified. See also *CONVERSION_ERROR Option [TSQL]*.

- **BYTE ORDER** specifies the byte order during reads. This option applies to all binary input fields. If none are defined, this option is ignored. SAP IQ always reads binary data in the format native to the machine it is running on (default is NATIVE). You can also specify:
    - HIGH when multibyte quantities have the high order byte first (for big endian platforms like Sun, IBM AIX, and HP).
    - LOW when multibyte quantities have the low order byte first (for little endian platforms like Windows).
- **LIMIT** specifies the maximum number of rows to insert into the table. The default is 0 for no limit. The maximum is 2^{31} - 1 (2147483647) rows.
- **NOTIFY** specifies that you be notified with a message each time the specified number of rows is successfully inserted into the table. The default is 0, meaning no notifications are printed. The value of this option overrides the value of the NOTIFY_MODULUS database option.
- **ON FILE ERROR** specifies the action SAP IQ takes when an input file cannot be opened because it does not exist or you have incorrect privileges to read the file. You can specify one of the following:
    - ROLLBACK – aborts the entire transaction (the default)
    - FINISH – finishes the insertions already completed and ends the load operation
    - CONTINUE – returns an error but only skips the file to continue the load operation

  Only one ON FILE ERROR clause is permitted.
- **PREVIEW** displays the layout of input into the destination table including starting position, name, and data type of each column. SAP IQ displays this information at the start of the load process. If you are writing to a log file, this information is also included in the log.
- **ROW DELIMITED BY delimiter-string** specifies a string up to 4 bytes in length that indicates the end of an input record. You can use this option only if all fields within the row are any of the following:
    - Delimited with column terminators
    - Data defined by the DATE or DATETIME `<column-spec>` options
    - ASCII fixed length fields

  Always include ROW DELIMITED BY to insure parallel loads. Omitting this clause from the LOAD specification may cause SAP IQ to load serially rather than in parallel.

  You cannot use this option if any input fields contain binary data. With this option, a row terminator causes any missing fields to be set to NULL. All rows must have the same row delimiters, and it must be distinct from all column delimiters. The row and field delimiter strings cannot be an initial subset of each other. For example, you cannot specify "*" as a field delimiter and "*#" as the row delimiter, but you could specify "#" as the field delimiter with that row delimiter.

If a row is missing its delimiters, SAP IQ returns an error and rolls back the entire load transaction. The only exception is the final record of a file where it rolls back that row and returns a warning message. On Windows, a row delimiter is usually indicated by the newline character followed by the carriage return character. You might need to specify this as the `<delimiter-string>` (see above for description) for either this option or FILLER.

- **SKIP** defines the number of rows to skip at the beginning of the input tables for this load. The maximum number of rows to skip is 2^{31} - 1 (2147483647). The default is 0. SKIP runs in single-threaded mode as it reads the rows to skip.
- **HEADER SKIP [ALL] ...HEADER DELIMITED BY** when you include `ALL` in your load statement for a multiple-file load, `LOAD TABLE` skips the number of header rows (that you specify with `<number>`) from the start of each file, while omitting `ALL` just skips the number of header rows from just the first file. `ALL` does not change the results for a single-file load.

  `HEADER SKIP <number>`, without `ALL`, specifies a number of lines at the beginning of the data file, including header rows, for `LOAD TABLE` to skip. All `LOAD TABLE` column specifications and other load options are ignored, until the specified number of rows is skipped.

  - The number of lines to skip is greater than or equal to zero.
  - Lines are determined by a 1-to-4-character delimiter string specified in the HEADER DELIMITED BY clause. The default HEADER DELIMITED BY string is the `\n` character.
  - The HEADER DELIMITED BY string has a maximum length of four characters. An error is returned, if the string length is greater than four or less than one.
  - When a non-zero HEADER SKIP value is specified, all data inclusive of the HEADER DELIMITED BY delimiter is ignored, until the delimiter is encountered the number of times specified in the HEADER SKIP clause.
  - All `LOAD TABLE` column specifications and other load options are ignored, until the specified number of rows has been skipped. After the specified number of rows has been skipped, the `LOAD TABLE` column specifications and other load options are applied to the remaining data.
  - The "header" bytes are ignored only at the beginning of the data. When multiple files are specified in the USING clause, HEADER SKIP only ignores data starting from the first row of the first file, until it skips the specified number of header rows, even if those rows exist in subsequent files. `LOAD TABLE` does not look for headers once it starts parsing actual data.
  - No error is reported, if `LOAD TABLE` processes all input data before skipping the number of rows specified by `HEADER SKIP`.

  `HEADER SKIP ALL` has the following behaviors:

  - You cannot specify `HEADER SKIP` and `HEADER SKIP ALL` in the same `LOAD TABLE` statement; doing so results in an error.
  - You can specify `HEADER SKIP ALL <number>` and `SKIP <number-of-rows>` together. When you do, the number of header rows (specified in `<number>`) is skipped first, then `SKIP <number-of-rows>` is performed until the statement reaches the number of rows you specified.
  - You can specify `HEADER SKIP ALL <number>` and `LIMIT <number-of-rows>`. When you do, the number of header rows (specified in `<number>`) is skipped first, then rows are loaded for the number of rows you specify.
- **WORD SKIP** allows the load to continue when it encounters data longer than the limit specified when the word index was created. If a row is not loaded because a word exceeds the maximum permitted size, a warning is written to the `.iqmsg` file. WORD size violations can be optionally logged to the MESSAGE LOG file and rejected rows logged to the ROW LOG file specified in the `LOAD TABLE` statement.

    - If the option is not specified, `LOAD TABLE` reports an error and rolls back on the first occurrence of a word that is longer than the specified limit.
    - `<number>` specifies the number of times the "`Words exceeding the maximum permitted word length not supported`" error is ignored.
    - 0 (zero) means there is no limit.
- **ON PARTIAL INPUT ROW** specifies the action to take when a partial input row is encountered during a load. You can specify one of the following:
    - CONTINUE – (default) issues a warning and continues the load operation.
    - ROLLBACK – aborts the entire load operation and reports this error:

      ```
      Partial input record skipped at EOF.
      SQLSTATE: QDC32    SQLSTATE: -1000232L
      ```
- **IGNORE CONSTRAINT** specifies whether to ignore CHECK, UNIQUE, NULL, DATA VALUE, and FOREIGN KEY integrity constraint violations that occur during a load and the maximum number of violations to ignore before initiating a rollback. Specifying each `<constrainttype>` has the following result:
    - CHECK `<limit>` – if `<limit>` specifies zero, the number of CHECK constraint violations to ignore is infinite. If CHECK is not specified, the first occurrence of any CHECK constraint violation causes the `LOAD` statement to roll back. If `<limit>` is nonzero, then the `<limit>` +1 occurrence of a CHECK constraint violation causes the load to roll back.
    - UNIQUE `<limit>` – if `<limit>` specifies zero, then the number of UNIQUE constraint violations to ignore is infinite. If `<limit>` is nonzero, then the `<limit>` +1 occurrence of a UNIQUE constraint violation causes the load to roll back.
    - NULL `<limit>` – if `<limit>` specifies zero, then the number of NULL constraint violations to ignore is infinite. If `<limit>` is nonzero, then the `<limit>` +1 occurrence of a NULL constraint violation causes the load to roll back.
    - FOREIGN KEY `<limit>` – if `<limit>` specifies zero, the number of FOREIGN KEY constraint violations to ignore is infinite. If `<limit>` is nonzero, then the `<limit>` +1 occurrence of a FOREIGN KEY constraint violation causes the load to roll back.
    - DATA VALUE `<limit>` – if the database option CONVERSION_ERROR = ON, an error is reported and the statement rolls back. If `<limit>` specifies zero, then the number of DATA VALUE constraint violations (data type conversion errors) to ignore is infinite. If `<limit>` is nonzero, then the `<limit>` +1 occurrence of a DATA VALUE constraint violation causes the load to roll back.
    - ALL `<limit>` – if the database option `CONVERSION_ERROR = ON`, an error is reported and the statement rolls back. If `<limit>` specifies zero, then the cumulative total of all integrity constraint violations to ignore is infinite. If `<limit>` is nonzero, then load rolls back when the cumulative total of all ignored UNIQUE, NULL, DATA VALUE, and FOREIGN KEY integrity constraint violations exceeds the value of `<limit>`. For example, you specify this IGNORE CONSTRAINT option:

      ```
      IGNORE CONSTRAINT NULL 50, UNIQUE 100, ALL 200
      ```

      The total number of integrity constraint violations cannot exceed 200, whereas the total number of NULL and UNIQUE constraint violations cannot exceed 50 and 100, respectively. Whenever any of these limits is exceeded, the LOAD TABLE statement rolls back.

      > **i Note**
      >
      > A single row can have more than one integrity constraint violation. Every occurrence of an integrity constraint violation counts towards the limit of that type of violation.

> Set the IGNORE CONSTRAINT option limit to a nonzero value if you are logging the ignored integrity constraint violations. Logging an excessive number of violations affects the performance of the load

If CHECK, UNIQUE, NULL, or FOREIGN KEY is not specified in the IGNORE CONSTRAINT clause, then the load rolls back on the first occurrence of each of these types of integrity constraint violation.

If DATA VALUE is not specified in the IGNORE CONSTRAINT clause, then the load rolls back on the first occurrence of this type of integrity constraint violation, unless the database option `CONVERSION_ERROR = OFF`. If `CONVERSION_ERROR = OFF`, a warning is reported for any DATA VALUE constraint violation and the load continues.

When the load completes, an informational message regarding integrity constraint violations is logged in the `.iqmsg` file. This message contains the number of integrity constraint violations that occurred during the load and the number of rows that were skipped.

- MESSAGE LOG specifies the names of files in which to log information about integrity constraint violations and the types of violations to log. Timestamps indicating the start and completion of the load are logged in both the MESSAGE LOG and the ROW LOG files. Both MESSAGE LOG and ROW LOG must be specified, or no information about integrity violations is logged.
    - If the ONLY LOG clause is not specified, no information on integrity constraint violations is logged. Only the timestamps indicating the start and completion of the load are logged.
    - Information is logged on all integrity constraint-type violations specified in the ONLY LOG clause or for all word index-length violations if the keyword WORD is specified.
    - If constraint violations are being logged, every occurrence of an integrity constraint violation generates exactly one row of information in the MESSAGE LOG file.
      The number of rows (errors reported) in the MESSAGE LOG file can exceed the IGNORE CONSTRAINT option limit, because the load is performed by multiple threads running in parallel. More than one thread might report that the number of constraint violations has exceeded the specified limit.
    - If constraint violations are being logged, exactly one row of information is logged in the ROW LOG file for a given row, regardless of the number of integrity constraint violations that occur on that row.
      The number of distinct errors in the MESSAGE LOG file might not exactly match the number of rows in the ROW LOG file. The difference in the number of rows is due to the parallel processing of the load described above for the MESSAGE LOG.
    - The MESSAGE LOG and ROW LOG files cannot be raw partitions or named pipes.
    - If the MESSAGE LOG or ROW LOG file already exists, new information is appended to the file.
    - Specifying an invalid file name for the MESSAGE LOG or ROW LOG file generates an error.
    - Specifying the same file name for the MESSAGE LOG and ROW LOG files generates an error.

  Various combinations of the IGNORE CONSTRAINT and MESSAGE LOG options result in different logging actions.

IGNORE CONSTRAINT Specified?	MESSAGE LOG Specified?	Action
Yes	Yes	All ignored integrity constraint violations are logged, including the user specified limit, before the rollback.
No	Yes	The first integrity constraint violation is logged before the rollback.
Yes	No	Nothing is logged.

IGNORE CONSTRAINT Specified?	MESSAGE LOG Specified?	Action
No	No	Nothing is logged. The first integrity constraint violation causes a rollback.

> → Tip
>
> Set the IGNORE CONSTRAINT option limit to a nonzero value, if you are logging the ignored integrity constraint violations. If a single row has more than one integrity constraint violation, a row for each violation is written to the MESSAGE LOG file. Logging an excessive number of violations affects the performance of the load.

- **LOG DELIMITED BY** specifies the separator between data values in the ROW LOG file. The default separator is a comma. SAP IQ no longer returns an error message when FORMAT BCP is specified as a `LOAD TABLE` clause. In addition, these conditions are verified and proper error messages are returned
  - If the specified load format is not ASCII, BINARY, or BCP, SAP IQ returns the message "`Only ASCII, BCP and BINARY are supported LOAD formats.`"
  - If the `LOAD TABLE` column specification contains anything other than column name, NULL, or ENCRYPTED, then SAP IQ returns the error message "`Invalid load specification for LOAD ... FORMAT BCP.`"
  - If the column delimiter or row terminator size for the FORMAT BCP load is greater than 10 characters, then SAP IQ returns the message "`Delimiter '%2' must be 1 to %3 characters in length.`" (where %3 equals 10).
    Messages corresponding to error or warning conditions which can occur for FORMAT BCP as well as FORMAT ASCII are the same for both formats.
  - If the load default value specified is AUTOINCREMENT, IDENTITY, or GLOBAL AUTOINCREMENT, SAP IQ returns the error "`Default value %2 cannot be used as a LOAD default value. %1`"
  - If the `LOAD TABLE` specification does not contain any columns that need to be loaded from the file specified, SAP IQ returns the error "`The LOAD statement must contain at least one column to be loaded from input file.`" and the `LOAD TABLE` statement rolls back.
  - If a load exceeds the limit on the maximum number of terms for a text document with TEXT indexes, SAP IQ returns the error "`Text document exceeds maximum number of terms. Support up to 4294967295 terms per document.`"

Examples

Example 1

load data from one file into the `Products` table on a Windows system. A tab is used as the column delimiter following the `Description` and `Color` columns:

```
LOAD TABLE Products
( ID ASCII(6),
FILLER(1),
Name   ASCII(15),
FILLER(1),
Description   '\x09',
Size   ASCII(2),
FILLER(1),
Color   '\x09',
Quantity   PREFIX 2,
UnitPrice   PREFIX 2,
FILLER(2) )
FROM 'C:\\mydata\\source1.dmp'
QUOTES OFF ESCAPES OFF
BYTE ORDER LOW
NOTIFY 1000
```

Example 2

load data from a file `a.inp` on a client computer:

```
LOAD TABLE t1(c1,c2,filler(30))
USING CLIENT FILE 'c:\\client-data\\a.inp'
QUOTES OFF ESCAPES OFF
IGNORE CONSTRAINT UNIQUE 0, NULL 0
MESSAGE LOG 'c:\\client-data\\m.log'
ROW LOG 'c:\\client-data\\r.log'ONLY LOG UNIQUE
```

Example 3

load data from two files into the `product_new` table (which allows NULL values) on a UNIX system. The tab character is the default column delimiter, and the newline character is the row delimiter:

```
LOAD TABLE product_new
( id,
name,
description,
size,
color   '\x09'   NULL( 'null', 'none', 'na' ),
quantity   PREFIX 2,
unit_price   PREFIX 2 )
FROM '/s1/mydata/source2.dump',
'/s1/mydata/source3.dump'
QUOTES OFF ESCAPES OFF
FORMAT ascii
DELIMITED BY '\x09'
ON FILE ERROR CONTINUE
ROW DELIMITED BY '\n'
```

Example 4

ignore 10 word-length violations; on the 11th, deploy the new error and roll back the load:

```
load table PTAB1(
       ck1         ','  null ('NULL') ,
       ck3fk2c2    ','  null ('NULL') ,
       ck4         ','  null ('NULL') ,
       ck5         ','  null ('NULL') ,
       ck6c1       ','  null ('NULL') ,
       ck6c2       ','  null ('NULL') ,
       rid         ','  null ('NULL')  )
FROM 'ri_index_selfRI.inp'
       row delimited by '\n'
       LIMIT 14   SKIP 10
       IGNORE CONSTRAINT UNIQUE 2, FOREIGN KEY 8
       word skip 10 quotes off escapes off strip
       off
…
```

Example 5

load data into table `t1` from the `BCP` character file `bcp_file.bcp` using the `FORMAT BCP` load option:

```
LOAD TABLE t1 (c1, c2, c3)
FROM 'bcp_file.bcp'
FORMAT BCP
…
```

Example 6

load default values 12345 into `c1` using the `DEFAULT` load option, and load `c2` and `c3` with data from the `LoadConst04.dat` file:

```
LOAD TABLE t1 (c1 DEFAULT '12345 ', c2, c3, filler(1))
FROM 'LoadConst04.dat'
STRIP OFF
QUOTES OFF ESCAPES OFF
DELIMITED BY ',';
```

Example 7

load `c1` and `c2` with data from the file `bcp_file.bcp` using the `FORMAT BCP` load option and set `c3` to the value 10:

```
LOAD TABLE t1 (c1, c2, c3 DEFAULT '10')
FROM 'bcp_file.bcp'
FORMAT BCP
QUOTES OFF ESCAPES OFF;
```

Example 8

this code fragment ignores one header row at the beginning of the data file, where the header row is delimited by `'&&'`:

```
LOAD TABLE
…HEADER SKIP 1 HEADER DELIMITED by '&&'
…
```

Example 9

this code fragment ignores 2 header rows at the beginning of the data file, where each header row is delimited by `'\n'`:

```
LOAD TABLE
…HEADER SKIP 2
…
```

Example 10

load data into RLV-enabled table `rvt1` from the `BCP` character file `bcp_file.bcp` using the `FORMAT BCP` load option:

```
LOAD TABLE rvt1 (c1, c2, c3)
FROM 'bcp_file.bcp'
FORMAT BCP
…
```

Example 11

load a table from a CSV file, using a double quotation mark for the QUOTE enclosure character and the backslash (\) for the QUOTE ESCAPE character:

```
LOAD TABLE tab1(c1, c2, c3)
    FROM 'foo.csv'
    DELIMITED BY ','
    ROW DELIMITED BY '\n'
    QUOTES ON ESCAPES OFF
    QUOTE '"'
    QUOTE ESCAPE '\'
    FORMAT CSV;
```

Example 12

If QUOTE ESCAPE is not specified and `foo.csv` file contains:

```
"""',"\\","\abc"
"\a\b\c","\\\\","""\dog"""
```

Executing the following LOAD TABLE statement:

```
LOAD TABLE tab1(c1,c2,c3)
        FROM 'foo.csv'
        DELIMITED BY ','
        ROW DELIMITED BY '\n'
        QUOTES ON ESCAPES OFF
        QUOTE '"'
        FORMAT CSV;
```

loads the data as:

```
c1 c2 c3
" \\ \abc
 \a\b\c \\\\ "dog"
```

Example 13

load a table, inserting the names of the files (`datafile1.csv` and `datafile2.csv`) into the `c2` column as specified by FILE NAME:

```
LOAD INTO TABLE test2 (c1, c2 FILE NAME, c3)
        USING FILE 'datafile1.csv', 'datafile2.csv'
        QUOTES OFF ESCAPES OFF FORMAT CSV
        DELIMITED BY ',' ROW DELIMITED BY '\n' ;
```

In this example, the contents of `datefile1.csv` are:

```
c1_val1, c3_value1,
c1_val2, c3_value2,
```

The contents of `datefile2.csv` are:

```
c1_val21, c3_value21,
c1_val22, c3_value22,
```

After the load, the `test2` table contains the following:

```
c1                c2                    c3
=========         ========              =========
c1_val1           datefile1.csv         c3_value1
c1_val2           datefile1.csv         c3_value2
c1_val21          datefile2.csv         c3_value21
c1_val22          datefile2.csv         c3_value22
```

Example 14

load a table named `test2` (which contains columns `c1` and `c2`, both of which are `VARCHAR(20)`), using files `datafile1.csv` and `datafile2.csv`, skipping the first header row from the start of each file:

```
LOAD INTO TABLE test2 (c1, c2 )
        USING FILE 'datafile1.csv', 'datafile2.csv'
        QUOTES OFF ESCAPES OFF FORMAT CSV
        HEADER SKIP ALL 1 DELIMITED BY ',' ROW DELIMITED BY '\n' ;
```

In this example, the contents of `datefile1.csv` are:

```
c1_val1, c3_value1,
c1_val2, c3_value2,
```

The contents of `datefile2.csv` are:

```
c1_val1, c3_value1,
c1_val2, c3_value2,
```

After the load, the contents of the `test2` table are:

```
c1                c2
=========         ======
c1_val2         c3_value2
c1_val22        c3_value22
```

Usage

The `LOAD TABLE` statement allows efficient mass insertion into a database table from a file with ASCII or binary data.

The `LOAD TABLE` options also let you control load behavior when integrity constraints are violated and to log information about the violations.

You can use `LOAD TABLE` on a temporary table, but the temporary table must have been declared with `ON COMMIT PRESERVE ROWS`, or the next `COMMIT` removes the rows you have loaded.

`LOAD TABLE` supports loading of large object (LOB) data.

SAP IQ supports loading from both ASCII and binary data, and it supports both fixed- and variable-length formats. To handle all of these formats, you must supply a `<load-specification>` to tell SAP IQ what kind of data to expect from each "column" or field in the source file. The `<column-spec>` lets you define these formats:

- ASCII with a fixed length of bytes. The `<input-width>` value is an integer indicating the fixed width in bytes of the input field in every record.
- Binary or non-binary fields that use a number of PREFIX bytes (1, 2, or 4) to specify the length of the input. There are two parts related to a `PREFIX` clause:

    - Prefix value – always a binary value.
    - Associated data bytes – always character format; never binary format.

  If the data is unloaded using the extraction facility with the `TEMP_EXTRACT_BINARY` option set ON, you must use the `BINARY WITH NULL BYTE` parameter for each column when you load the binary data.
- Variable-length characters delimited by a separator. You can specify the terminator as hexadecimal ASCII characters. The `<delimiter-string>` can be any string of up to 4 characters, including any combination of printable characters, and any 8-bit hexadecimal ASCII code that represents a nonprinting character. For example, specify:
    - '\x09' to represent a tab as the terminator.
    - '\x00' for a null terminator (no visible terminator as in "C" strings).
    - '\x0a' for a newline character as the terminator. You can also use the special character combination of '\n' for newline.

  > i Note
  >
  > The delimiter string can be from 1 to 4 characters long, but you can specify only a single character in the `DELIMITED BY` clause. For `BCP`, the delimiter can be up to 10 characters.

- DATE or DATETIME string as ASCII characters. You must define the `<input-date-format>` or `<input-datetime-format>` of the string using one of the corresponding formats for the date and datetime data types supported by SAP IQ. Use `DATE` for date values and `DATETIME` for datetime and time values.

Option	Meaning
yyyy or YYYY yy or YY	Represents number of year. Default is current year.
mm or MM	Represents number of month. Always use leading zero or blank for number of the month where appropriate, for example, '05' for May. `DATE` value must include a month. For example, if the `DATE` value you enter is 1998, you receive an error. If you enter '03', SAP IQ applies the default year and day and converts it to '1998-03-01'.
dd or DD jjj or JJJ	Represents number of day. Default day is 01. Always use leading zeros for number of day where appropriate, for example, '01' for first day. J or j indicates a Julian day (1 to 366) of the year.
hh HH	Represents hour. Hour is based on 24-hour clock. Always use leading zeros or blanks for hour where appropriate, for example, '01' for 1 am. '00' is also valid value for hour of 12 a.m.
nn	Represents minute. Always use leading zeros for minute where appropriate, for example, '08' for 8 minutes.
ss[.ssssss]	Represents seconds and fraction of a second.
aa	Represents the a.m. or p.m. designation.
pp	Represents the p.m. designation only if needed. (This is an incompatibility with SAP IQ versions earlier than 12.0; previously, "pp" was synonymous with "aa".)
hh	SAP IQ assumes zero for minutes and seconds. For example, if the `DATETIME` value you enter is '03', SAP IQ converts it to '03:00:00.0000'.

Option	Meaning
hh:nn or hh:mm	SAP IQ assumes zero for seconds. For example, if the time value you enter is '03:25', SAP IQ converts it to '03:25:00.0000'.

Table 12: Sample DATE and DATETIME Format Options

Input Data	Format Specification
12/31/98	DATE ('MM/DD/YY')
19981231	DATE ('YYYYMMDD')
123198140150	DATETIME ('MMDDYYhhnnss')
14:01:50 12-31-98	DATETIME ('hh:nn:ss MM-DD-YY')
18:27:53	DATETIME ('hh:nn:ss')
12/31/98 02:01:50AM	DATETIME ('MM/DD/YY hh:nn:ssaa')

- The `FILE NAME` option allows you to insert the name of a file into a column when you perform a LOAD INTO TABLE statement. When you do, the name of the file (but not its contents) is loaded into the column for each of the rows in the table.
  - You can only specify this option for VARCHAR and CHAR columns.
  - The length of the file name cannot be longer than the maximum length of the column you are specifying.
  - You can only specify one column for use with `FILE NAME`.
  - After the load inserts the file name into a column, the system does not add any information to mark the column as a FILE NAME column.
  - The column you specify for FILE NAME cannot contain any data.

SAP IQ has built-in load optimizations for common date, time, and datetime formats. If your data to be loaded matches one of these formats, you can significantly decrease load time by using the appropriate format.

You can also specify the date/time field as an ASCII fixed-width field (as described above) and use the FILLER(1) option to skip the column delimiter.

The NULL portion of the `<column-spec>` indicates how to treat certain input values as NULL values when loading into the table column. These characters can include BLANKS, ZEROS, or any other list of literals you define. When specifying a NULL value or reading a NULL value from the source file, the destination column must be able to contain NULLs.

`ZEROS` are interpreted as follows: the cell is set to NULL if (and only if) the input data (before conversion, if ASCII) is all binary zeros (and not character zeros).

- If the input data is character zero, then:
  1. NULL (ZEROS) never causes the cell to be NULL.
  2. NULL ('0') causes the cell to be NULL.
- If the input data is binary zero (all bits clear), then:
  1. NULL (ZEROS) causes the cell to be NULL.
  2. NULL ('0') never causes the cell to be NULL.

For example, if your `LOAD` statement includes `col1 date('yymmdd') null(zeros)` and the date is 000000, you receive an error indicating that 000000 cannot be converted to a `DATE`(4). To get `LOAD TABLE` to insert a NULL value in `col1` when the data is 000000, either write the NULL clause as `null('000000')`, or modify the data to equal binary zeros and use NULL(ZEROS).

If the length of a VARCHAR cell is zero and the cell is not NULL, you get a zero-length cell. For all other data types, if the length of the cell is zero, SAP IQ inserts a NULL. This is ANSI behavior. For non-ANSI treatment of zero-length character data, set the NON_ANSI_NULL_VARCHAR database option.

Use the DEFAULT option to specify a load default column value. You can load a default value into a column, even if the column does not have a default value defined in the table schema. This feature provides more flexibility at load time.

- The LOAD TABLE DEFAULTS option must be ON in order to use the default value specified in the LOAD TABLE statement. If the DEFAULTS option is OFF, the specified load default value is not used and a NULL value is inserted into the column instead.
- The LOAD TABLE command must contain at least one column that needs to be loaded from the file specified in the LOAD TABLE command. Otherwise, an error is reported and the load is not performed.
- The specified load default value must conform to the supported default values for columns and default value restrictions. The LOAD TABLE DEFAULT option does not support AUTOINCREMENT, IDENTITY, or GLOBAL AUTOINCREMENT as a load default value.
- The LOAD TABLE DEFAULT <default-value> must be of the same character set as that of the database.
- Encryption of the default value is not supported for the load default values specified in the LOAD TABLE DEFAULT clause.
- A constraint violation caused by evaluation of the specified load default value is counted for each row that is inserted in the table.

Another important part of the <load-specification> is the FILLER option. This option indicates you want to skip over a specified field in the source input file. For example, there may be characters at the end of rows or even entire fields in the input files that you do not want to add to the table. As with the <column-spec> definition, FILLER specifies ASCII fixed length of bytes, variable length characters delimited by a separator, and binary fields using PREFIX bytes.

Standards

- SQL—Vendor extension to ISO/ANSI SQL grammar.
- SAP Database Products—Not applicable.

Permissions

The privileges required to execute a LOAD TABLE statement depend on the database server -gl command line option, as follows:

-gl ALL (default) you must have one of:

- You are the owner of the table
- ALTER object-level privilege on the table
- LOAD object-level privilege on the table
- ALTER ANY TABLE system privilege
- LOAD ANY TABLE system privilege

- ALTER ANY OBJECT system privilege

-gl DBA you must have one of these system privileges:

- ALTER ANY TABLE
- LOAD ANY TABLE
- ALTER ANY OBJECT

-gl NONE execution of the `LOAD TABLE` statement is not permitted.

For more information on the `-gl` command line option, please refer *SAP IQ Utility Guide > start_iq Database Server Startup Utility > start_iq Server Options*.

`LOAD TABLE` also requires a write lock on the table.

When using the USING CLIENT FILE clause:

- READ CLIENT FILE system privilege is also required.
- Read privileges are required on the directory being read from.
- The `ALLOW_READ_CLIENT_FILE` database option must be enabled.
- The `ALLOW_READ_CLIENT_FILE` secure feature must be enabled.

3.1.7 String Comparisons on Encrypted Text

If data is case-insensitive, or uses a collation other than ISO_BINENG, you must decrypt ciphertext columns to perform string comparisons.

When performing comparisons on strings, the distinction between equal and identical strings is important for many collations and depends on the `CASE` option of `CREATE DATABASE`. In a database that is set to `CASE RESPECT` and uses the ISO_BINENG collation, the defaults for SAP IQ, equality, and identity questions are resolved the same way.

Identical strings are always equal, but equal strings may not be identical. Strings are identical only if they are represented using the same byte values. When data is case-insensitive or uses a collation where multiple characters must be treated as equal, the distinction between equality and identity is significant. ISO1LATIN1 is such a collation.

For example, the strings “ABC” and “abc” in a case-insensitive database are not identical but are equal. In a case-sensitive database, they are neither identical nor equal.

The ciphertext created by the Sybase encryption functions preserves identity but not equality. In other words, the ciphertext for “ABC” and “abc” will never be equal.

To perform equality comparisons on ciphertext when your collation or `CASE` setting does not allow this type of comparison, your application must modify the values within that column into some canonical form, where there are no equal values that are not also identical values. For example, if your database is created with `CASE IGNORE` and the ISO_BINENG collation and your application applies UCASE to all input values before placing them into the column, then all equal values are also identical.

3.1.8 Database Options for Column Encryption

Certain SAP IQ database option settings affect column encryption and decryption; the default settings are not optimal for most column encryption operations.

In this section:

Protect Ciphertext from Accidental Truncation [page 271]
: To prevent accidental truncation of the ciphertext output of the encrypt function (or accidental truncation of any other character or binary string), set the STRING_RTRUNCATION database option.

Preserve Ciphertext Integrity [page 271]
: Set ASE_BINARY_DISPLAY to preserve ciphertext integrity.

Prevent Misuse of Ciphertext [page 272]
: Set CONVERSION_MODE to prevent implicit data type conversions of encrypted data that result in semantically meaningless operations.

3.1.8.1 Protect Ciphertext from Accidental Truncation

To prevent accidental truncation of the ciphertext output of the encrypt function (or accidental truncation of any other character or binary string), set the STRING_RTRUNCATION database option.

```
SET OPTION STRING_RTRUNCATION = 'ON'
```

When STRING_RTRUNCATION is ON (the default), the engine raises an error whenever a string would be truncated during a load, insert, update, or SELECT INTO operation. This is ISO/ANSI SQL behavior and is a recommended practice.

When explicit truncation is required, use a string expression such as LEFT, SUBSTRING, or CAST.

Setting STRING_RTRUNCATION OFF forces silent truncation of strings.

The AES_DECRYPT function also checks input ciphertext for valid data length, and checks text output to verify both the resulting data length and the correctness of the supplied key. If you supply the data type argument, the data type is checked as well.

3.1.8.2 Preserve Ciphertext Integrity

Set ASE_BINARY_DISPLAY to preserve ciphertext integrity.

```
SET OPTION ASE_BINARY_DISPLAY = 'OFF'
```

When ASE_BINARY_DISPLAY is OFF (the default), the system leaves binary data unmodified, and in its raw binary form.

When ASE_BINARY_DISPLAY is ON, the system converts binary data into its hexadecimal string display representation. Temporarily set the option ON only if you need to show data to an end user, or if you need to export the data to another external system, where raw binary may become altered in transit.

3.1.8.3 Prevent Misuse of Ciphertext

Set `CONVERSION_MODE` to prevent implicit data type conversions of encrypted data that result in semantically meaningless operations.

The `CONVERSION_MODE` database option restricts implicit conversion between binary data types (`BINARY`, `VARBINARY`, and `LONG BINARY`) and other nonbinary data types (`BIT`, `TINYINT`, `SMALLINT`, `INT`, `UNSIGNED INT`, `BIGINT`, `UNSIGNED BIGINT`, `CHAR`, `VARCHAR`, and `LONG VARCHAR`) on various operations:

```
SET TEMPORARY OPTION CONVERSION_MODE = 1
```

Setting `CONVERSION_MODE` to 1 restricts implicit conversion of binary data types to any other nonbinary data type on `INSERT` and `UPDATE` commands, and in queries. The restrict binary conversion mode also applies to `LOAD TABLE` default values and `CHECK` constraint.

The `CONVERSION_MODE` option default value of 0 maintains the implicit conversion behavior of binary data types in versions of SAP IQ earlier than 12.7.

In this section:

CONVERSION_MODE Option [page 272]
Restricts implicit conversion between binary data types (`BINARY`, `VARBINARY`, and `LONG BINARY`) and other non-binary data types (`BIT`, `TINYINT`, `SMALLINT`, `INT`, `UNSIGNED INT`, `BIGINT`, `UNSIGNED BIGINT`, `CHAR`, `VARCHAR`, and `LONG VARCHAR`) on various operations. Also allows all explicit conversions to be permitted as implicit conversions on various operations.

3.1.8.3.1 CONVERSION_MODE Option

Restricts implicit conversion between binary data types (`BINARY`, `VARBINARY`, and `LONG BINARY`) and other non-binary data types (`BIT`, `TINYINT`, `SMALLINT`, `INT`, `UNSIGNED INT`, `BIGINT`, `UNSIGNED BIGINT`, `CHAR`, `VARCHAR`, and `LONG VARCHAR`) on various operations. Also allows all explicit conversions to be permitted as implicit conversions on various operations.

Allowed Values

0, 1, 2

Default

0

Scope

Option can be set at the database (PUBLIC) or user level. At the database level, the value becomes the default for any new user, but has no impact on existing users. At the user level, overrides the PUBLIC value for that user only. No system privilege is required to set option for self. System privilege is required to set at database level or at user level for any user other than self.

Requires the SET ANY PUBLIC OPTION system privilege to set this option. Can be set temporary for an individual connection or for the PUBLIC role. Takes effect immediately.

Remarks

The default `CONVERSION_MODE` value of 0 maintains implicit conversion behavior prior to version 12.7.

Setting `CONVERSION_MODE` to 1 restricts implicit conversion of binary data types to any other non-binary data type on `INSERT`, `UPDATE`, and in queries. The restrict binary conversion mode also applies to `LOAD TABLE` default values and `CHECK` constraint. `CONVERSION_MODE 1` prevents implicit data type conversions of encrypted data that would result in semantically meaningless operations.

Setting `CONVERSION_MODE` option to allows all explicit conversions to be permitted as implicit conversions on various operations. If this option is not set, the user must use `CAST` or `CONVERT` in queries that require explicit conversions.

Users must be specifically licensed to use the encrypted column functionality of the SAP IQ Advanced Security Option.

Implicit Conversion Restrictions

The `CONVERSION_MODE` option restrict binary mode value of 1 (`CONVERSION_MODE` = 1) restricts implicit conversion for these operations:

- `LOAD TABLE` with `CHECK` constraint or default value
- `INSERT...SELECT`, `INSERT...VALUE`, and `INSERT...LOCATION`
- Certain types of `UPDATE`
- Certain types of `INSERT` and `UPDATE` via updatable cursor
- All aspects of queries in general

3.1.9 Encryption and Decryption Example

An example using the `AES_ENCRYPT` and `AES_DECRYPT` functions, written in commented SQL.

```
--   This example of aes_encrypt and aes_decrypt function use is presented in
three parts:
--
```

```
-- Part I: Preliminary description of target tables and users as DDL
-- Part II: Example schema changes motivated by introduction of encryption
-- Part III: Use of views and stored procedures to protect encryption keys
--
--   Part I: Define target tables and users
--   Assume two classes of user, represented here by the instances
--   PrivUser and NonPrivUser, assigned to groups reflecting differing
--   privileges.
--   The initial state reflects the schema prior to the introduction
--   of encryption.
--   Set up the starting context: There are two tables with a common key.
--   Some columns contain sensitive data, the remaining columns do not.
--   The usual join column for these tables is sensitiveA.
--   There is a key and a unique index.
    grant connect to PrivUser identified by 'verytrusted' ;
    grant connect to NonPrivUser identified by 'lesstrusted' ;
    grant connect to high_privileges_group ;
    create role high_privileges_group ;
    grant role high_privileges_group to PrivUser ;
    grant connect to low_privileges_group ;
    create role low_privileges_group ;
    grant role low_privileges_group to NonPrivUser ;
    create table DBA.first_table
                    (sensitiveA char(16) primary key
                    ,sensitiveB numeric(10,0)
                    ,publicC    varchar(255)
                    ,publicD    date
                    )  ;
--   There is an implicit unique HG (High_Group) index enforcing the primary key.
    create table second_table
                    (sensitiveA char(16)
                    ,publicP integer
                    ,publicQ tinyint
                    ,publicR varchar(64)
                    )  ;
    create hg index second_A_HG on second_table ( sensitiveA ) ;
--   TRUSTED users can see the sensitive columns.
    grant select ( sensitiveA, sensitiveB, publicC, publicD )
        on DBA.first_table to PrivUser ;
    grant select ( sensitiveA, publicP, publicQ, publicR )
        on DBA.second_table to PrivUser ;
--   Non-TRUSTED users in existing schema need to see sensitiveA to be
--   able to do joins, even though they should not see sensitiveB.
    grant select ( sensitiveA, publicC, publicD )
        on DBA.first_table to NonPrivUser ;
    grant select ( sensitiveA, publicP, publicQ, publicR )
        on DBA.second_table to NonPrivUser ;
--   Non-TRUSTED users can execute queries such as
    select I.publicC, 3*II.publicQ+1
    from DBA.first_table I, DBA.second_table II
    where I.sensitiveA = II.sensitiveA and I.publicD IN ( '2006-01-11' ) ;
--   and
    select count(*)
    from DBA.first_table I, DBA.second_table II
    where I.sensitiveA = II.sensitiveA and SUBSTR(I.sensitiveA,4,3)
    BETWEEN '345' AND '456' ;
--   But only TRUSTED users can execute the query
    select I.sensitiveB, 3*II.publicQ+1
    from DBA.first_table I, DBA.second_table II
    where I.sensitiveA = II.sensitiveA and I.publicD IN ( '2006-01-11' ) ;
--   Part II: Change the schema in preparation for encryption
--
--    The DBA introduces encryption as follows:
--
--    For applicable tables, the DBA changes the schema, adjusts access
--    permissions, and updates existing data. The encryption
--    keys used are hidden in a subsequent step.
--   DataLength comparison for length of varbinary encryption result
--   (units are Bytes):
--
--    PlainText CipherText    Corresponding Numeric Precisions
--
--            0       16
--      1 -  16       32      numeric(1,0)   - numeric(20,0)
--     17 -  32       48      numeric(21,0)  - numeric(52,0)
--     33 -  48       64      numeric(53,0)  - numeric(84,0)
--     49 -  64       80      numeric(85,0)  - numeric(116,0)
--     65 -  80       96      numeric(117,0) - numeric(128,0)
--     81 -  96      112
--     97 - 112      128
--    113 - 128      144
--    129 - 144      160
--    145 - 160      176
--    161 - 176      192
--    177 - 192      208
--    193 - 208      224
--    209 - 224      240
--   The integer data types tinyint, small int, integer, and bigint
--   are varbinary(32) ciphertext.
--   The exact relationship is
--   DATALENGTH(ciphertext) =
--   (((DATALENGTH(plaintext)+ 15) / 16) + 1) * 16
--   For the first table, the DBA chooses to preserve both the plaintext and
--   ciphertext forms. This is not typical and should only be done if the
--   database files are also encrypted.
--   Take away NonPrivUser's access to column sensitiveA and transfer
--   access to the ciphertext version.
--   Put a unique index on the ciphertext column. The ciphertext
--   itself is indexed.
--   NonPrivUser can select the ciphertext and use it.
--   PrivUser can still select either form (without paying decrypt costs).
    revoke select ( sensitiveA ) on DBA.first_table from NonPrivUser ;
    alter table DBA.first_table add encryptedA varbinary(32) ;
    grant select ( encryptedA ) on DBA.first_table to PrivUser ;
    grant select ( encryptedA ) on DBA.first_table to NonPrivUser ;
    create unique hg index first_A_unique on first_table ( encryptedA ) ;
    update DBA.first_table
        set encryptedA = aes_encrypt(sensitiveA, 'seCr3t')
        where encryptedA is null ;
    commit
--   Now change column sensitiveB.
    alter table DBA.first_table add encryptedB varbinary(32) ;
    grant select ( encryptedB ) on DBA.first_table to PrivUser ;
    create unique hg index first_B_unique on first_table ( encryptedB ) ;
    update DBA.first_table
        set encryptedB = aes_encrypt(sensitiveB,
        'givethiskeytonoone') where encryptedB is null ;
    commit
--   For the second table, the DBA chooses to keep only the ciphertext.
--   This is more typical and encrypting the database files is not required.
    revoke select ( sensitiveA ) on DBA.second_table from NonPrivUser ;
    revoke select ( sensitiveA ) on DBA.second_table from PrivUser ;
    alter table DBA.second_table add encryptedA varbinary(32) ;
    grant select ( encryptedA ) on DBA.second_table to PrivUser ;
    grant select ( encryptedA ) on DBA.second_table to NonPrivUser ;
    create unique hg index second_A_unique on second_table ( encryptedA ) ;
    update DBA.second_table
        set encryptedA = aes_encrypt(sensitiveA, 'seCr3t')
        where encryptedA is null ;
    commit
    alter table DBA.second_table drop sensitiveA ;
--   The following types of queries are permitted at this point, before
--   changes are made for key protection:
--   Non-TRUSTED users can equi-join on ciphertext; they can also select
--   the binary, but have no way to interpret it.
    select I.publicC, 3*II.publicQ+1
    from DBA.first_table I, DBA.second_table II
    where I.encryptedA = II.encryptedA and I.publicD IN ( '2006-01-11' ) ;
--   Ciphertext-only access rules out general predicates and expressions.
--   The following query does not return meaningful results.
--
--   NOTE: These four predicates can be used on the varbinary containing
--   ciphertext:
--      = (equality)
--      <> (inequality)
--      IS NULL
--      IS NOT NULL
    select count(*)
    from DBA.first_table I, DBA.second_table II
    where I.encryptedA = II.encryptedA and SUBSTR(I.encryptedA,4,3)
        BETWEEN '345' AND '456' ;
--   The TRUSTED user still has access to the plaintext columns that
--   were retained. Therefore, this user does not need to call
--   aes_decrypt and does not need the key.
    select count(*)
    from DBA.first_table I, DBA.second_table II
    where I.encryptedA = II.encryptedA and SUBSTR(I.sensitiveA,4,3)
        BETWEEN '345' AND '456' ;
--   Part III: Protect the encryption keys
--   This section illustrates how to grant access to the plaintext, but
--   still protect the keys.
--   For the first table, the DBA elected to retain the plaintext columns.
--   Therefore, the following view has the same capabilities as the trusted
--   user above.
--   Assume group_member is being used for additional access control.
--   NOTE: In this example, NonPrivUser still has access to the ciphertext
--   encrypted in the base table.
    create view DBA.a_first_view (sensitiveA, publicC, publicD)
        as
            select
              IF group_member('high_privileges_group',user_name()) = 1
                 THEN sensitiveA
                 ELSE NULL
              ENDIF,
              publicC,
              publicD
            from first_table ;
    grant select on DBA.a_first_view to PrivUser ;
    grant select on DBA.a_first_view to NonPrivUser ;
--   For the second table, the DBA did not keep the plaintext.
--   Therefore, aes_decrypt calls must be used in the view.
--   IMPORTANT: Hide the view definition with ALTER VIEW, so that no one
--   can discover the key.
    create view DBA.a_second_view (sensitiveA,publicP,publicQ,publicR)
        as
            select
              IF group_member('high_privileges_group',user_name()) = 1
                 THEN aes_decrypt(encryptedA,'seCr3t', char(16))
                 ELSE NULL
              ENDIF,
              publicP,
              publicQ,
              publicR
            from second_table ;
    alter view DBA.a_second_view set hidden ;
    grant select on DBA.a_second_view to PrivUser ;
    grant select on DBA.a_second_view to NonPrivUser ;
--   Likewise, the key used for loading can be protected in a stored
     procedure.
--   By hiding the procedure (just as the view is hidden), no-one can see
--   the keys.
    create procedure load_first_proc(@inputFileName varchar(255),
                        @colDelim varchar(4) default '$',
                        @rowDelim varchar(4) default '\n')
        begin
            execute immediate with quotes
                'load table DBA.second_table
                (encryptedA encrypted(char(16),' ||
                '''' || 'seCr3t' || '''' || '),publicP,publicQ,publicR) ' ||
                ' from ' || '''' || @inputFileName || '''' ||
                ' delimited by ' || '''' || @colDelim || '''' ||
                ' row delimited by ' || '''' || @rowDelim || '''' ||
                ' quotes off escapes off' ;
        end
    ;
    alter procedure DBA.load_first_proc set hidden ;
--   Call the load procedure using the following syntax:
    call load_first_proc('/dev/null', '$', '\n') ;
--   Below is a comparison of several techniques for protecting the
--   encryption keys by using user-defined functions (UDFs), other views,
--   or both. The first and the last alternatives offer maximum performance.
--   The second_table is secured as defined earlier.
--   Alternative 1:
--   This baseline approach relies on restricting access to the entire view.
     create view
            DBA.second_baseline_view(sensitiveA,publicP,publicQ,publicR)
        as
            select
              IF group_member('high_privileges_group',user_name()) = 1
                 THEN aes_decrypt(encryptedA,'seCr3t', char(16))
                 ELSE NULL
              ENDIF,
              publicP,
              publicQ,
              publicR
            from DBA.second_table ;
     alter view DBA.second_baseline_view set hidden ;
     grant select on DBA.second_baseline_view to NonPrivUser ;
     grant select on DBA.second_baseline_view to PrivUser ;
--   Alternative 2:
--     Place the encryption function invocation within a user-defined
--   function (UDF).
--    Hide the definition of the UDF. Restrict the UDF permissions.
--    Use the UDF in a view that handles the remainder of the security
--     and business logic.
--    Note: The view itself does not need to be hidden.
     create function DBA.second_decrypt_function(IN datum varbinary(32))
            RETURNS char(16) DETERMINISTIC
            BEGIN
                RETURN aes_decrypt(datum,'seCr3t', char(16));
            END  ;
     grant execute on DBA.second_decrypt_function to PrivUser ;
     alter function DBA.second_decrypt_function set hidden ;
     create view
            DBA.second_decrypt_view(sensitiveA,publicP,publicQ,publicR)
        as
                select
                  IF group_member('high_privileges_group',user_name()) = 1
                     THEN second_decrypt_function(encryptedA)
                     ELSE NULL
                  ENDIF,
                  publicP,
                 publicQ,
                 publicR
                from DBA.second_table ;
     grant select on DBA.second_decrypt_view to NonPrivUser ;
     grant select on DBA.second_decrypt_view to PrivUser ;
--  Alternative 3:
--    Sequester only the key selection in a user-defined function.
--    This function could be extended to support selection of any
--  number of keys.
--    This UDF is also hidden and has restricted execute privileges.
--    Note: Any view that uses this UDF therefore does not compromise
-- the key values.
     create function DBA.second_key_function()
              RETURNS varchar(32) DETERMINISTIC
              BEGIN
                return 'seCr3t' ;
              END
     grant execute on DBA.second_key_function to PrivUser ;
     alter function DBA.second_key_function set hidden ;
     create view DBA.second_key_view(sensitiveA,publicP,publicQ,publicR)
                as
                    select
                      IF group_member('high_privileges_group',user_name()) = 1
                         THEN aes_decrypt(encryptedA,second_key_function(),
                         char(16))
                         ELSE NULL
                      ENDIF,
                      publicP,
                      publicQ,
                      publicR
                    from DBA.second_table ;
     grant select on DBA.second_key_view to NonPrivUser ;
     grant select on DBA.second_key_view to PrivUser ;
--  Alternative 4:
--    The recommended alternative is to separate the security logic
--    from the business logic by dividing the concerns into two views.
--    Only the security logic view needs to be hidden.
--    Note: The performance of this approach is similar to that of the first
-- alternative.
     create view
        DBA.second_SecurityLogic_view(sensitiveA,publicP,publicQ,publicR)
            as
                select
                  IF group_member('high_privileges_group',user_name()) = 1
                     THEN aes_decrypt(encryptedA,'seCr3t', char(16))
                     ELSE NULL
                  ENDIF,
                  publicP,
                  publicQ,
                  publicR
                from DBA.second_table ;
     alter view DBA.second_SecurityLogic_view set hidden ;
     create view
        DBA.second_BusinessLogic_view(sensitiveA,publicP,publicQ,publicR)
            as
                select
                  sensitiveA,
                  publicP,
                  publicQ,
                  publicR
                from DBA.second_SecurityLogic_view ;
     grant select on DBA.second_BusinessLogic_view to NonPrivUser ;
     grant select on DBA.second_BusinessLogic_view to PrivUser ;
-- End of encryption example
```

Related Information

AES_ENCRYPT Function [String] [page 245]

AES_DECRYPT Function [String] [page 249]

LOAD TABLE ENCRYPTED Clause [page 250]

3.2 Kerberos Authentication Support in SAP IQ

SAP IQ supports Kerberos authentication, a login feature that allows you to maintain a single user ID and password for both database connections and operating system and network logins.

You can use your Kerberos credentials to connect to the database without specifying a user ID or password.

Kerberos authentication is part of the separately licensed SAP IQ Advanced Security Option.

In this section:

3.2.1 Licensing Requirements for Kerberos

The Advanced Security Option (IQ_SECURITY) protects your environment against unauthorized access, and is required to use Kerberos authentication with SAP IQ.

3.3 LDAP User Authentication Support in SAP IQ

You can integrate SAP IQ into any existing enterprise-wide directory access framework based on Lightweight Directory Access Protocol (LDAP), a widely accepted international standard.

In this section:

3.3.1 License Requirements for LDAP User Authentication

The Advanced Security Option (IQ_SECURITY) protects your environment against unauthorized access, and is required to allow LDAP user authentication with SAP IQ.

4 Appendix: SQL Reference

Reference material for SQL statements, database options, functions, and system procedures mentioned in this document.

In this section:

4.1 SQL Statements

Interactive SQL statements customize and modify the database.

In this section:

DROP LDAP SERVER Statement [page 311]

Removes the named LDAP server configuration object from the SYSLDAPSERVER system view after verifying that the LDAP server configuration object is not in a READY or ACTIVE state.

DROP LOGIN POLICY Statement [page 312]

Removes a login policy from the database.

DROP ROLE Statement [page 313]

Removes a user-defined role from the database or converts a user-extended role to a regular user.

DROP USER Statement [page 315]

Removes a user.

GRANT CHANGE PASSWORD Statement [page 316]

Allows users to manage passwords for other users and administer the CHANGE PASSWORD system privilege.

GRANT CONNECT Statement [page 318]

Grants CONNECT privilege to a user.

GRANT CREATE Statement [page 320]

Grants CREATE privilege on a specified dbspace to the specified users and roles.

GRANT EXECUTE Statement [page 321]

Grants EXECUTE privilege on a procedure or user-defined function.

GRANT Object-Level Privilege Statement [page 321]

Grants database object-level privileges on individual tables or views to a user or role.

GRANT ROLE Statement [page 323]

Grants roles to users or other roles, with or without administrative rights.

GRANT SET USER Statement [page 327]

Grants the ability for one user to impersonate another user and to administer the SET USER system privilege.

GRANT System Privilege Statement [page 329]

Grants specific system privileges to users or roles, with or without administrative rights.

GRANT USAGE ON SEQUENCE Statement [page 333]

Grants the USAGE system privilege on a specified sequence to a user or role.

REVOKE CHANGE PASSWORD Statement [page 333]

Removes the ability of a user to manage passwords and administer the system privilege.

REVOKE CONNECT Statement [page 335]

Removes a user from the database.

REVOKE CREATE Statement [page 336]

Removes CREATE privileges on the specified dbspace from the specified user IDs.

REVOKE EXECUTE Statement [page 337]

Removes EXECUTE permissions that were given using the GRANT statement.

REVOKE Object-Level Privilege Statement [page 338]

Removes object-level privileges that were given using the GRANT statement.

REVOKE ROLE Statement [page 340]

Removes a users membership in a role or his or her ability to administer the role.

REVOKE SET USER Statement [page 342]

Removes the ability for one user to impersonate another user and to administer the SET USER system privilege.

REVOKE System Privilege Statement [page 344]
Removes specific system privileges from specific users and the right to administer the privilege.

REVOKE USAGE ON SEQUENCE Statement [page 347]
Removes USAGE privilege on a specified sequence.

SET OPTION Statement [page 348]
Changes options that affect the behavior of the database and its compatibility with Transact-SQL. Setting the value of an option can change the behavior for all users or an individual user, in either a temporary or permanent scope.

SETUSER Statement [page 351]
Allows a user to temporarily assume the roles and system privileges of another user (also known as impersonation) to perform operations, provided they already have the minimum required privileges to perform the task to begin with.

VALIDATE LDAP SERVER Statement [page 353]
Validates changes to the settings of existing LDAP server configuration objects before applying them.

4.1.1 ALTER LDAP SERVER Statement

Any changes to an LDAP server configuration object are applied on subsequent connections. Any connection already started when the change is applied does not immediately reflect the change.

Syntax

```
ALTER LDAP SERVER <ldapua-server-name>
   { ldapua-server-attribs
   | [ WITH  ( SUSPEND | ACTIVATE | REFRESH ) ] }
ldapua-server-attribs -
   SEARCH DN
      URL { <'URL_string' | NULL> }
      | ACCESS ACCOUNT { <'DN_string' | NULL> }
      | IDENTIFIED BY ( <'password' | NULL> }
      | IDENTIFIED BY ENCRYPTED { <encrypted-password | NULL> }
   | AUTHENTICATION URL { <'URL_string' | NULL> }
   | CONNECTION TIMEOUT <timeout_value>
   | CONNECTION RETRIES <retry_value>
   | TLS { ON | OFF }
```

Parameters

- **URL** Identifies the host (by name or by IP address), port number, and the search to be performed for the DN lookup for a given user ID. This value is validated for correct LDAP URL syntax before it is stored in the ISYSLDAPSERVER system table. The maximum size for this string is 1024 bytes.

- **ACCESS ACCOUNT** User created in the LDAP server for use by SAP IQ, not a user within SAP IQ. The distinguished name (DN) for this user is used to connect to the LDAP server. This user has permissions within the LDAP server to search for DNs by user ID in the locations specified by the SEARCH DN URL. The maximum size for this string is 1024 bytes.
- **IDENTIFIED BY** Provides the password associated with the ACCESS ACCOUNT user. The password is stored using symmetric encryption on disk. Use the value NULL to clear the password and set it to none. The maximum size of a clear text password is 255 bytes.
- **IDENTIFIED BY ENCRYPTED** Configures the password associated with the ACCESS ACCOUNT distinguished name in an encrypted format. The binary value is the encrypted password and is stored on disk as is. Use the value NULL to clear the password and set it to none. The maximum size of the binary is 289 bytes. The encrypted key should be a valid varbinary value. Do not enclose the encrypted key in quotation marks.
- **AUTHENTICATION URL** Identifies the host (by name or IP address) and the port number of the LDAP server to use for authentication of the user. This is the value defined for URL_string and is validated for correct LDAP URL syntax before it is stored in `ISYSLDAPSERVER` system table. The DN of the user obtained from a prior DN search and the user password bind a new connection to the authentication URL. A successful connection to the LDAP server is considered proof of the identity of the connecting user. The maximum size for this string is 1024 bytes.
- **CONNECTION TIMEOUT** Specifies the connection timeout from SAP IQ to the LDAP server for both DN searches and authentication. This value is in milliseconds, with a default value of 10 seconds.
- **CONNECTION RETRIES** Specifies the number of retries on connections from SAP IQ to the LDAP server for both DN searches and authentication. The valid range of values is 1– 60, with a default value of 3.
- **TLS** Defines whether the TLS or Secure LDAP protocol is used for connections to the LDAP server for both DN searches and authentication. When set to ON, the TLS protocol is used and the URL would being with "ldap://" When set to OFF (or not specified), Secure LDAP protocol is used and the URL begins with “ldaps://”. When using the TLS protocol, specify the database security option TRUSTED_CERTIFICATES_FILE with a file name containing the certificate of the Certificate Authority (CA) that signed the certificate used by the LDAP server.
- **WITH ACTIVATE** Activates the LDAP server configuration object for immediate use upon creation. This permits the definition and activation of LDAP User Authentication in one statement. The LDAP server configuration object state changes to READY when WITH ACTIVATE is used.

Examples

Example 1

suspends the LDAP server configuration object named `apps_primary`:

```
ALTER LDAP SERVER apps_primary SUSPEND
```

Example 2

changes the LDAP server configuration object named `apps_primary` to use a different URL for authentication on host `fairfax`, sets the port number to `1066`, sets the number of connection retries to `10`, and finally activates the LDAP server configuration object:

```
ALTER LDAP SERVER apps_primary
AUTHENTICATION URL 'ldap://my_LDAPserver:1066/'
CONNECTION RETRIES 10
```

```
WITH ACTIVATE
```

Usage

In addition to resetting LDAP server configuration object values for attributes, the `ALTER LDAP SERVER` statement allows an administrator to make manual adjustments to a server's state and behavior by putting the LDAP server configuration object in maintenance mode and returning it to service from maintenance mode.

Standards

ANSI SQL–Compliance level: Transact-SQL extension.

Permissions

Requires the MANAGE ANY LDAP SERVER system privilege.

4.1.2 ALTER LOGIN POLICY Statement

Changes existing login policies or configures logical server access.

Syntax

Syntax 1

```
ALTER LOGIN POLICY <policy-name>
   { { ADD | DROP | SET } LOGICAL SERVER ls-assignment-list
   [ LOGICAL SERVER ls-override-list ])
ls-assignment-list -
   { { ls-name, ...}
   | ALL
   | COORDINATOR
   | SERVER
   | NONE
   | DEFAULT }
ls-override-list -
   { ls-name, ...}
ls-name -
   { OPEN | <user-defined-ls-name> }
```

Syntax 2

```
ALTER LOGIN POLICY <policy-name> policy-option
```

```
policy-option -
   policy-option-name = policy-option-value
policy-option-name -
  AUTO_UNLOCK_TIME
    | CHANGE_PASSWORD_DUAL_CONTROL
    | DEFAULT_LOGICAL_SERVER
    | LOCKED
    | MAX_CONNECTIONS
    | MAX_DAYS_SINCE_LOGIN
    | MAX_FAILED_LOGIN_ATTEMPTS
    | MAX_NON_DBA_CONNECTIONS
    | PAM_FAILOVER_TO_STD
    | PAM_SERVICENAME
    | PASSWORD_EXPIRY_ON_NEXT_LOGIN
    | PASSWORD_GRACE_TIME
    | PASSWORD_LIFE_TIME
    | ROOT_AUTO_UNLOCK_TIME
    | LDAP_PRIMARY_SERVER
    | LDAP_SECONDARY_SERVER
    | LDAP_AUTO_FAILBACK_PERIOD
    | LDAP_FAILOVER_TO_STD
    | LDAP_REFRESH_DN
policy-option-value -
    { UNLIMITED | DEFAULT | <value> }
```

Parameters

- **policy-name** The name of the login policy. Specify root to modify the root login policy.
- **policy-option-value** The value assigned to the login policy option. If you specify UNLIMITED, no limits are used. If you specify DEFAULT, the default limits are used. See *Login Policy Options* and *LDAP Login Policy Options* for supported values for each option.
- **policy-option-name** The name of the policy option. See *Login Policy Options* and *LDAP Login Policy Options* for details about each option.

Applies to

Simplex and multiplex.

Examples

Example 1

sets the password_life_time value to `UNLIMITED` and the max_failed_login_attempts value to `5` in the `Test1` login policy:

```
ALTER LOGIN POLICY Test1
password_life_time=UNLIMITED
max_failed_login_attempts=5;
```

Usage

If you do not specify a policy option, values for this login policy come from the root login policy. New policies do not inherit the MAX_NON_DBA_CONNECTIONS and ROOT_AUTO_UNLOCK_TIME policy options.

All new databases include a root login policy. You can modify the root login policy values, but you cannot delete the policy.

Permissions

Requires the MANAGE ANY LOGIN POLICY system privilege.

In this section:

4.1.2.1 Login Policy Options

Available options for root and user-defined login policies.

Option	Description
AUTO_UNLOCK_TIME	The time period after which locked accounts that are not granted the MANAGE ANY USER system privilege are automatically unlocked. You can define this option in any login policy, including the root login policy. **Values** 0 – UNLIMITED **Default** UNLIMITED **Applies to** all users who are not granted the MANAGE ANY USER system privilege.
CHANGE_PASSWORD_DUAL_CONTROL	Requires input from two users, each of whom is granted the CHANGE PASSWORD system privilege, to change the password of another user. **Values** ON, OFF **Default** OFF **Applies to** all users.

Option	Description
DEFAULT_LOGICAL_SERVER	If the connection string specifies no logical server, the user connects to the DEFAULT_LOGICAL_SERVER option specified in the user's login policy. **Values** • Name of an existing user-defined logical server • AUTO – value of the default logical server in the root login policy. • COORDINATOR – the current coordinator node. • NONE – denies access to any multiplex server. • OPEN – use alone or with the name of a user-defined logical server. Allows access to all multiplex nodes that are not members of any user-defined logical servers. • SERVER – allows access to all of the multiplex nodes, subject to the semantics of the SERVER logical server. **Default** AUTO **Applies to** all users. Requires MANAGE MULTIPLEX system privilege.
LOCKED	If set ON, users cannot establish new connections. This setting temporarily denies access to login policy users. Logical server overrides for this option are not allowed. **Values** ON, OFF **Default** OFF **Applies to** all users except those with the MANAGE ANY USER system privilege.
MAX_CONNECTIONS	The maximum number of concurrent connections allowed for a user. You can specify a per-logical-server setting for this option. **Values** 0 – 2147483647 **Default** UNLIMITED **Applies to** all users except those with the SERVER OPERATOR or DROP CONNECTION system privilege.
MAX_DAYS_SINCE_LOGIN	The maximum number of days that can elapse between two successive logins by the same user. **Values** 0 – 2147483647 **Default** UNLIMITED **Applies to** all users except those with the MANAGE ANY USER system privilege.
MAX_FAILED_LOGIN_ATTEMPTS	The maximum number of failed attempts, since the last successful attempt, to log in to the user account before the account is locked. **Values** 0 – 2147483647 **Default** UNLIMITED **Applies to** all users.

Option	Description
MAX_NON_DBA_CONNECTIONS	The maximum number of concurrent connections that a user without SERVER OPERATOR or DROP CONNECTION system privileges can make. This option is supported only in the root login policy. **Values** 0 – 2147483647 **Default** UNLIMITED **Applies to** all users except those with the SERVER OPERATOR or DROP CONNECTION privilege.
PAM_FAILOVER_TO_STD	Use standard authentication if PAM authentication is enabled but the PAM library is unavailable due to a system failure. Authentication failures returned by PAM do not fail over to standard authentication. **Values** ON, OFF **Default** ON **Applies to** all users.
PAM_SERVICE_NAME	The PAM service name to use when authenticating. The service name identifies the rule set to be used by PAM during validation. If empty (the default), do not use PAM. The database server continues to function when PAM support is unavailable. See *Enabling PAM User Authentication* in *Administration: User Management and Security*.
PASSWORD_EXPIRY_ON_NEXT_LOGIN	If set ON, the user's password expires at the next login. **Values** ON, OFF **Default** OFF **Applies to** all users. ℹ Note This functionality is not currently implemented when logging in to SAP IQ Cockpit. However, when logging in to SAP IQ outside of SAP IQ Cockpit (for example, using Interactive SQL), users are then prompted to enter a new password.
PASSWORD_GRACE_TIME	The number of days before password expiration during which login is allowed but the default post_login procedure issues warnings. **Values** 0 – 2147483647 **Default** 0 **Applies to** all users.
PASSWORD_LIFE_TIME	The maximum number of days before a password must be changed. **Values** 0 – 2147483647 **Default** UNLIMITED **Applies to** all users.

Option	Description
ROOT_AUTO_UNLOCK_TIME	The time period after which locked accounts that are granted the MANAGE ANY USER system privilege are automatically unlocked. You can define this option only in the root login policy. **Values** 0 – UNLIMITED **Default** 15 **Applies to** all users who are granted the MANAGE ANY USER system privilege.

4.1.2.2 LDAP Login Policy Options

Available login policy options for LDAP user authentication

Option	Description
LDAP_PRIMARY_SERVER	Specifies the name of the primary LDAP server. **Values** n/a **Default** None **Applies to** All users.
LDAP_SECONDARY_SERVER	Specifies the name of the secondary LDAP server. **Values** n/a **Default** None **Applies to** All users.
LDAP_AUTO_FAILBACK_PERIOD	Specifies the time period, in minutes, after which automatic failback to the primary server is attempted. **Values** 0 - 2147483647 **Default** 15 minutes **Applies to** All users.
LDAP_FAILOVER_TO_STD	Permits authentication with standard authentication when authentication with the LDAP server fails due to system resources, network outage, connection timeouts, or similar system failures. However, it does not permit an actual authentication failure returned from an LDAP server to fail over to standard authentication. **Values** ON, OFF **Default** ON **Applies to** All users.

Option	Description
LDAP_REFRESH_DN	Updates the ldap_refresh_dn value in the ISYSLOGINPOLICYOPTION system table with the current time, stored in Coordinated Universal Time (UTC). Each time a user authenticates with LDAP, if the value of ldap_refresh_dn in ISYSLOGINPOLICYOPTION is more recent than the value of user_dn in ISYSUSER, a search for a new user DN occurs. The user_dn value is then updated with the new user DN and the user_dn_changed_at value is again updated to the current time. **Values** NOW **Initial value for ROOT policy** NULL **Initial value for user-defined login policy** Current time stored in UTC **Applies to** All users.

4.1.2.3 Multiplex Login Policy Configuration

Configure login policies for multiplex servers.

Example

This example overrides the login policy settings on a logical server, increasing the maximum number of connections on logical server ls1:

```
ALTER LOGIN POLICY lp1 max_connections=20 LOGICAL SERVER ls1;
```

Usage

Applies only to multiplex.

Any login management commands you execute on any multiplex server automatically propagate to all servers in the multiplex. For best performance, execute these commands, or any DDL, on the coordinator.

An override at the logical server level override means that a particular login policy option has different settings for different logical servers. SYS.ISYSIQLSLOGINPOLICYOPTION stores login policy option values for logical-server override. For each logical-server override of a login policy option, a corresponding row exists in ISYSIQLSLOGINPOLICYOPTION.

4.1.2.4 Logical Server Access Configuration

Configure logical server access.

Example

Assume that the root login policy allows access to logical servers `ls4` and `ls5` and login policy `lp1` exists with no logical server assignment. The statement below effectively assigns login policy `lp1` to logical servers `ls4` and `ls5`.

Assign logical server `ls1` to login policy `lp1`:

```
ALTER LOGIN POLICY lp1 ADD LOGICAL SERVER ls1
```

Example

This statement allows access of logical servers `ls2` and `ls3` from login policy `lp1`:

```
ALTER LOGIN POLICY lp1 ADD LOGICAL SERVER ls2, ls3
```

Example

Modify login policy `lp1` to allow access to `ls3` and `ls4`only:

```
ALTER LOGIN POLICY lp1 ADD LOGICAL SERVER ls4
ALTER LOGIN POLICY lp1 DROP LOGICAL SERVER ls1, ls2
```

or:

```
ALTER LOGIN POLICY lp1 SET LOGICAL SERVER ls3, ls4
```

Example

Modify login policy`lp1` to deny access to any logical servers:

```
ALTER LOGIN POLICY lp1 SET LOGICAL SERVER NONE
```

Example

Drop current logical server assignments of login policylp1 and allow it to inherit the logical server assignments of the root login policy:

```
ALTER LOGIN POLICY lp1 SET LOGICAL SERVER DEFAULT
```

Usage

ADD, DROP, or SET clauses let you configure the logical server assignments of a login policy:

ADD adds new logical server assignments to a login policy.
DROP deletes existing logical server assignments from a login policy.
SET replaces all logical server assignments for a login policy with a new set of logical server.

Use only one ADD, DROP, or SET clause. Use SERVER, NONE, and DEFAULT clauses only with the SET clause. Specify a particular logical server name only once per ls-assignment list or ls-override list.

An error is returned if:

- Any logical server specified with the ADD clause is already assigned to the login policy.
- Any logical server specified with the DROP clause is currently not assigned to the login policy.
- Logical server assignment change may cause a membership overlap among assigned logical servers.

SYS.ISYSIQLOGINPOLICYLSINFO stores logical server assignment information. For each logical-server override of a login policy option, a corresponding row exists in ISYSIQLOGINPOLICYLSINFO.

4.1.3 ALTER ROLE Statement

Migrates a compatibility role to a user-defined system role, then automatically drops the compatibility role.

> i Note
>
> You cannot use the ALTER ROLE statement to migrate SYS_AUTH_SA_ROLE or SYS_AUTH_SSO_ROLE. These roles are automatically migrated when SYS_AUTH_DBA_ROLE is migrated.

Syntax

Syntax 1 – To migrate SYS_AUTH_DBA_ROLE

```
ALTER ROLE <predefined_sys_role_name>
   MIGRATE TO <new_role_name [, new_sa_role_name, new_sso_role_name]>
```

Syntax 2 – To migrate all other compatibility roles

```
ALTER ROLE <predefined_sys_role_name>
   MIGRATE TO <new_role_name>
```

Parameters

- **predefined_sys_role_name** the name of a compatibility role that still exists (has not already been dropped) in the database.
- **new_role_name** the name of the new role cannot begin with the prefix SYS_ or end with the suffix _ROLE.
- **new_sa_role_name** required only when migrating SYS_AUTH_DBA_ROLE. The new role to which the underlying system privileges of SYS_AUTH_SA_ROLE are to be migrated to cannot already exist in the database, and the new role name cannot begin with the prefix SYS_ or end with the suffix _ROLE.
- **new_sso_role_name** required only when migrating SYS_AUTH_DBA_ROLE. The new role to which the underlying system privileges of SYS_AUTH_SSO_ROLE are to be migrated to cannot already exist in the database, and the new role name cannot begin with the prefix SYS_ or end with the suffix _ROLE.

Examples

Example 1

migrates SYS_AUTH_DBA_ROLE to the new roles `Custom_DBA`, `Custom_SA`, and `Custom_SSO` respectively. It then automatically migrates all users, underlying system privileges, and roles granted to SYS_AUTH_DBA_ROLE to the applicable new roles. Finally, it drops SYS_AUTH_DBA_ROLE, SYS_AUTH_SA_ROLE, and SYS_AUTH_SSO_ROLE.

```
ALTER ROLE SYS_AUTH_DBA_ROLE
MIGRATE TO Custom_DBA, Custom_SA, Custom_SSO
```

Example 2

migrates SYS_AUTH_OPERATOR_ROLE role to the new role `Operator_role`. It then automatically migrates all users, underlying system privileges, and roles granted to SYS_AUTH_OPERATOR_ROLE to the new role and drops SYS_AUTH_OPERATOR_ROLE.

```
ALTER ROLE SYS_AUTH_OPERATOR_ROLE
MIGRATE TO Operator_role
```

Usage

During the migration process:

- A new user-defined role is created.
- All of the system privileges currently granted to the migrating predefined role are automatically granted to the new user-defined role.

- All users and roles currently granted to the migrating predefined role are automatically granted to the new user-defined role.
- The compatibility role is dropped.

Since no role administrator was specified during the migration process, only global role administrators can manage the new role. Use the CREATE ROLE statement to add role administrators with appropriate administrative rights to the role.

Standards

ANSI SQL–Compliance level: Transact-SQL extension.

Permissions

Requires the MANAGE ROLES system privilege granted with administrative rights.

4.1.4 ALTER USER Statement

Changes user settings.

Syntax

Syntax 1

Change the definition of a database user

```
ALTER USER <user-name>
   | [ IDENTIFIED BY <password> ]
   | [ LOGIN POLICY <policy-name> ]
   | [ FORCE PASSWORD CHANGE { ON | OFF } ]
```

Syntax 2

Refresh the Distinguished Name (DN) for an LDAP user

```
ALTER USER <user-name>
   REFRESH DN
```

Syntax 3

Revert a user's login policy to the original values

```
ALTER USER <user-name>
   RESET LOGIN POLICY
```

Syntax 4

Change a user's password when CHANGE_PASSWORD_DUAL_CONTROL is enabled in a user's login policy

```
ALTER USER <user-name>
   IDENTIFIED [ FIRST | LAST ] BY <password_part>
```

Parameters

- **user-name** Name of the user.
- **IDENTIFIED BY** The password for the user. Clause is not supported (ERROR) when CHANGE_PASSWORD_DUAL_CONTROL option is enabled in a user's login policy
- **IDENTIFIED[FIRST | LAST] BY** Clause mandatory when CHANGE_PASSWORD_DUAL_CONTROL option is enabled in a target user's login policy. FIRST | LAST keyword specifies the part of the dual password part being defined.
- **policy-name** Name of the login policy to assign the user. No change is made if you do not specify a login policy. No change is made if the LOGIN POLICY clause is not specified.
- **FORCE PASSWORD CHANGE** Controls whether the user must specify a new password upon logging in. This setting overrides the PASSWORD_EXPIRY_ON_NEXT_LOGIN option setting in the user's login policy.

  > i Note
  >
  > This functionality is not currently implemented when logging in to SAP IQ Cockpit. However, when logging in to SAP IQ outside of SAP IQ Cockpit (for example, using Interactive SQL), users are then prompted to enter a new password.

- **password** You do not have to specify a password for the user. A user without a password cannot connect to the database. This is useful if you are creating a role and do not want anyone to connect to the database using the role user ID. A user ID must be a valid identifier. User IDs and passwords cannot:
  - Begin with white space, single quotes, or double quotes
  - End with white space
  - Contain semicolons

  A password can be either a valid identifier, or a string (maximum 255 characters) placed in single quotes. Passwords are case-sensitive. The password should be composed of 7-bit ASCII characters, as other characters may not work correctly if the database server cannot convert them from the client's character set to UTF-8.

  You can use the VERIFY_PASSWORD_FUNCTION option to specify a function to implement password rules (for example, passwords must include at least one digit). If you do use a password verification function, you cannot specify more than one user ID and password in the `GRANT CONNECT` statement.

  The encryption algorithm used for hashing the user passwords is FIPS-certified encryption support:
  - The DLL is called dbfips10.dll.
  - The HASH function accepts the algorithms: SHA1_FIPS SHA256_FIPS.
  - If the -fips server option is specified and an algorithm that is not FIPS-certified is given to the HASH function, the database server uses SHA1_FIPS instead of SHA1, SHA256_FIPS instead of SHA256, and returns an error if MD5 is used (MD5 is not a FIPS-certified algorithm).
  - If the -fips option is specified, the database server uses SHA256_FIPS for password hashing.

- **RESET LOGIN POLICY** Reverts the settings of the user's login to the original values in the login policy. This usually clears all locks that are implicitly set due to the user exceeding the failed logins or exceeding the maximum number of days since the last login. When you reset a login policy, a user can access an account that has been locked for exceeding a login policy option limit such as MAX_FAILED_LOGIN_ATTEMPTS or MAX_DAYS_SINCE_LOGIN.
- **REFRESH DN** Clears the saved DN and timestamp for a user, which is used during LDAP authentication.

Examples

Example 1

alters a user named `SQLTester`. The password is set to `welcome`. The `SQLTester` user is assigned to the `Test1` login policy and the password does not expire on the next login:

```
ALTER USER SQLTester
IDENTIFIED BY welcome
LOGIN POLICY Test1
FORCE PASSWORD CHANGE OFF
```

Example 2

clears the distinguished name (DN) and timestamp for a user named `Mary` used for LDAP authentication:

```
ALTER USER Mary REFRESH DN
```

Example 3

sets the password for user3 to PassPart1PassPart2. This assumes that `user1` and `user2` have the CHANGE PASSWORD system privilege and the change_password_dual_control option is enabled (ON) in the login policy for `user3`:

User1 enters:

```
ALTER USER user3 IDENTIFIED FIRST BY PassPart1
```

User2 enters:

```
ALTER USER user3 IDENTIFIED LAST BY PassPart2
```

Once set, user3 logs on by entering the password `PassPart1PassPart2`.

Usage

User IDs and passwords cannot:

- Begin with white space, single quotes, or double quotes
- End with white space
- Contain semicolons

Passwords cannot exceed 255 characters.

If you set the `PASSWORD_EXPIRY_ON_NEXT_LOGIN` value to ON, the passwords of all users assigned to this login policy expire immediately when he or she next logs in. You can use the `ALTER USER` and `LOGIN POLICY` clauses to force users to change their passwords at the next login.

If the CHANGE_PASSWORD_DUAL CONTROL login policy option is disable (OFF) during the dual password change process:

- the target user will be unable to log in with the single password part already defined. The `ALTER USER` command must be reissued using single password control syntax.
- If the option is disabled after the dual password change process is complete, but before the target user logs in, there is no impact on the target user. The target user must log in using both password parts.

If the target user is already logged in when the dual password change process occurs, the user cannot change their password in the current session until both parts of the new password are set. Once the dual password change process is complete, the target user can use `GRANT CONNECT`, `ALTER USER`, `sp_password`, or `sp_iqpassword` to the password without first logging out. The prompt to enter the current password, use the new dual control password, not the password originally entered for the current session.

The `GRANT CONNECT` statement is not supported during for the dual password change process to set either password part. However, once the dual password change process is complete, the target user can use the `GRANT CONNECT` statement, `ALTER USER`, `sp_password`, or `sp_iqpassword` to change their password without first logging out.

As soon as both parts of the password are successfully specified by users with the CHANGE PASSWORD system privilege, the password for the target user is automatically expired. This forces the target user to change the password the next time he or she logs in.

The encryption algorithm used for hashing the user passwords is FIPS-certified encryption support:

- The DLL is called `dbfips10.dll`
- The HASH function accepts the algorithms: `SHA1_FIPS` `SHA256_FIPS`
- If the `-fips` server option is specified and an algorithm that is not FIPS-certified is given to the HASH function, the database server uses `SHA1_FIPS` instead of `SHA1`, `SHA256_FIPS` instead of `SHA256`, and returns an error if `MD5` is used (`MD5` is not a FIPS-certified algorithm).
- If the `-fips` option is specified, the database server uses SHA256_FIPS for password hashing.

Standards

- SQL–Vendor extension to ISO/ANSI SQL grammar.
- SAP Database Products–Not supported by SAP ASE.

Permissions

- To change own password – None required.
- To change the password of any user – Requires the CHANGE PASSWORD system privilege.
- To use the `LOGIN POLICY`, `FORCE PASSWORD CHANGE`, `RESET LOGIN POLICY`, or `REFRESH DN` clauses requires the MANAGE ANY USER system privilege.

4.1.5 CREATE LDAP SERVER Statement

Creates a new LDAP server configuration object for LDAP user authentication. Parameters defined during the creation of an LDAP server configuration object are stored in the ISYSLDAPSERVER (system view SYSLDAPSERVER) system table.

Syntax

```
CREATE LDAP SERVER <ldapua-server-name>
   [ ldapua-server-attribs ]
   [ WITH ACTIVATE ]
ldapua-server-attribs
   SEARCH DN
      URL { <'URL_string' | NULL> }
      | ACCESS ACCOUNT { <'DN_string' | NULL> }
      | IDENTIFIED BY ( <'password' | NULL> }
      | IDENTIFIED BY ENCRYPTED { <encrypted-password | NULL> }
   | AUTHENTICATION URL { <'URL_string' | NULL> }
   | CONNECTION TIMEOUT <timeout_value>
   | CONNECTION RETRIES <retry_value>
   | TLS { ON | OFF }
```

Parameters

- **URL** Identifies the host (by name or by IP address), port number, and the search to be performed for the DN lookup for a given user ID. This value is validated for correct LDAP URL syntax before it is stored in the ISYSLDAPSERVER system table. The maximum size for this string is 1024 bytes.
- **ACCESS ACCOUNT** User created in the LDAP server for use by SAP IQ, not a user within SAP IQ. The distinguished name (DN) for this user is used to connect to the LDAP server. This user has permissions within the LDAP server to search for DNs by user ID in the locations specified by the SEARCH DN URL. The maximum size for this string is 1024 bytes.
- **IDENTIFIED BY** Provides the password associated with the ACCESS ACCOUNT user. The password is stored using symmetric encryption on disk. Use the value NULL to clear the password and set it to none. The maximum size of a clear text password is 255 bytes.
- **IDENTIFIED BY ENCRYPTED** Configures the password associated with the ACCESS ACCOUNT distinguished name in an encrypted format. The binary value is the encrypted password and is stored on disk as is. Use the value NULL to clear the password and set it to none. The maximum size of the binary is 289 bytes. The encrypted key should be a valid varbinary value. Do not enclose the encrypted key in quotation marks.
- **AUTHENTICATION URL** Identifies the host (by name or IP address) and the port number of the LDAP server to use for authentication of the user. This is the value defined for URL_string and is validated for correct LDAP URL syntax before it is stored in ISYSLDAPSERVER system table. The DN of the user obtained from a prior DN search and the user password bind a new connection to the authentication URL. A successful connection to the LDAP server is considered proof of the identity of the connecting user. The maximum size for this string is 1024 bytes.
- **CONNECTION TIMEOUT** Specifies the connection timeout from SAP IQ to the LDAP server for both DN searches and authentication. This value is in milliseconds, with a default value of 10 seconds.

- **CONNECTION RETRIES** Specifies the number of retries on connections from SAP IQ to the LDAP server for both DN searches and authentication. The valid range of values is 1– 60, with a default value of 3.
- **TLS** Defines whether the TLS or Secure LDAP protocol is used for connections to the LDAP server for both DN searches and authentication. When set to ON, the TLS protocol is used and the URL would being with "ldap://" When set to OFF (or not specified), Secure LDAP protocol is used and the URL begins with “ldaps://”. When using the TLS protocol, specify the database security option TRUSTED_CERTIFICATES_FILE with a file name containing the certificate of the Certificate Authority (CA) that signed the certificate used by the LDAP server.
- **WITH ACTIVATE** Activates the LDAP server configuration object for immediate use upon creation. This permits the definition and activation of LDAP User Authentication in one statement. The LDAP server configuration object state changes to READY when WITH ACTIVATE is used.

Examples

Example 1

sets the search parameters, the authentication URL, and sets a three second timeout, and activates the server so it can begin authenticating users. It connects to the LDAP server without TLS or SECURE LDAP protocols.

```
SET OPTION PUBLIC.login_mode = 'Standard,LDAPUA'
CREATE LDAP SERVER apps_primary
SEARCH DN
    URL 'ldap://my_LDAPserver:389/dc=MyCompany,dc=com??sub?cn=*'
    ACCESS ACCOUNT 'cn=aseadmin, cn=Users, dc=mycompany, dc=com'
    IDENTIFIED BY 'Secret99Password'
AUTHENTICATION URL 'ldap://my_LDAPserver:389/'
CONNECTION TIMEOUT 3000
WITH ACTIVATE
```

Example 2

uses the same search parameters as example 1, but specifies "ldaps" so that a Secure LDAP connection is established with the LDAP server on host my_LDAPserver, port 636. Only LDAP clients using the Secure LDAP protocol may now connect on this port. The database security option TRUSTED_CERTIFICATE_FILE must be set with a file name containing the certificate of the certificate authority (CA) that signed the certificate used by the LDAP server at "ldaps://my_LDAPserver:636". During the handshake with the LDAP server, the certificate presented by the LDAP server is checked by the SAP IQ server (the LDAP client) to ensure that it is signed by one of the certificates listed in the file. This establishes trust by the client that the server is who it says it is. The ACCESS ACCOUNT and IDENTIFIED BY parameters establish trust by the LDAP server that the client is who it says it is.

> ℹ Note
>
> The TLS parameter must be OFF when Secure LDAP is used instead of TLS protocol.

```
SET OPTION PUBLIC.login_mode = 'Standard,LDAPUA'
SET OPTION PUBLIC.trusted_certificates_file = '/mycompany/shared/trusted.txt'
CREATE LDAP SERVER secure_primary
SEARCH DN
    URL  'ldaps://my_LDPAserver:636/dc=MyCompany,dc=com??sub?cn=*'
    ACCESS ACCOUNT 'cn=aseadmin, cn=Users, dc=mycompany, dc=com'
    IDENTIFIED BY 'Secret99Password'
AUTHENTICATION URL 'ldaps://my_LDAPserver:636/'
```

```
CONNECTION TIMEOUT 3000
TLS OFF
WITH ACTIVATE
```

Example 3

establishes the TLS protocol on port 389. It also requires database security option TRUSTED_CERTIFICATE_FILE to be set with a file name and provides the same type of security as example 2. In this example, the TLS protocol is ON to facilitate wider support by LDAP server vendors.

> **i Note**
>
> Check the requirements of all your LDAP servers when deciding how to configure Secure LDAP or TLS for an SAP IQ server.

```
SET OPTION PUBLIC.login_mode = 'Standard,LDAPUA'
SET OPTION PUBLIC.trusted_certificates_file = '/mycompany/shared/trusted.txt'
CREATE LDAP SERVER tls_primary
SEARCH DN
    URL  'ldap://my_LDAPserver:389/dc=MyCompany,dc=com??sub?cn=*'
    ACCESS ACCOUNT 'cn=aseadmin, cn=Users, dc=mycompany, dc=com'
    IDENTIFIED BY 'Secret99Password'
AUTHENTICATION URL 'ldap://my_LDAPserver:389/'
CONNECTION TIMEOUT 3000
TLS ON
WITH ACTIVATE
```

Standards

ANSI SQL–Compliance level: Transact-SQL extension.

Permissions

Requires the MANAGE ANY LDAP SERVER system privilege.

4.1.6 CREATE LOGIN POLICY Statement

Creates a login policy in the database.

Syntax

```
CREATE LOGIN POLICY <policy-name> policy-option
policy-option
   policy-option-name = policy-option-value
policy-option-name
```

```
   AUTO_UNLOCK_TIME
    | CHANGE_PASSWORD_DUAL_CONTROL
    | DEFAULT_LOGICAL_SERVER
    | LOCKED
    | MAX_CONNECTIONS
    | MAX_DAYS_SINCE_LOGIN
    | MAX_FAILED_LOGIN_ATTEMPTS
    | MAX_NON_DBA_CONNECTIONS
    | PAM_FAILOVER_TO_STD
    | PAM_SERVICENAME
    | PASSWORD_EXPIRY_ON_NEXT_LOGIN
    | PASSWORD_GRACE_TIME
    | PASSWORD_LIFE_TIME
    | ROOT_AUTO_UNLOCK_TIME
    | LDAP_PRIMARY_SERVER
    | LDAP_SECONDARY_SERVER
    | LDAP_AUTO_FAILBACK_PERIOD
    | LDAP_FAILOVER_TO_STD
    | LDAP_REFRESH_DN
policy-option-value
    { UNLIMITED | DEFAULT | <value> }
```

Parameters

- **policy-name** The name of the login policy. Specify root to modify the root login policy.
- **policy-option-name** The name of the policy option. See *Login Policy Options* and *LDAP Login Policy Options* for details about each option.
- **policy-option-value** The value assigned to the login policy option. If you specify UNLIMITED, no limits are used. If you specify DEFAULT, the default limits are used. See *Login Policy Options* and *LDAP Login Policy Options* for supported values for each option.

Applies to

Simplex and multiplex.

Examples

Example 1

creates the `Test1` login policy. This login policy has an unlimited password life and allows the user a maximum of five attempts to enter a correct password before the account is locked.

```
CREATE LOGIN POLICY Test1
password_life_time=UNLIMITED
max_failed_login_attempts=5;
```

Usage

If you do not specify a policy option, values for this login policy come from the root login policy. New policies do not inherit the MAX_NON_DBA_CONNECTIONS and ROOT_AUTO_UNLOCK_TIME policy options.

Permissions

Requires MANAGE ANY LOGIN POLICY system privilege.

The following system privileges can override the noted login policy options:

Exception System Privilege	Login Policy Option
SERVER OPERATOR or DROP CONNECTION system privilege	MAX_NON_DBA_CONNS MAX_CONNECTIONS
MANAGE ANY USER system privilege	LOCKED MAX_DAYS_SINCE_LOGIN

In this section:

4.1.6.1 Login Policy Options

Available options for root and user-defined login policies.

Option	Description
AUTO_UNLOCK_TIME	The time period after which locked accounts that are not granted the MANAGE ANY USER system privilege are automatically unlocked. You can define this option in any login policy, including the root login policy. **Values** 0 – UNLIMITED **Default** UNLIMITED **Applies to** all users who are not granted the MANAGE ANY USER system privilege.

Option	Description
CHANGE_PASSWORD_DUAL_CONTROL	Requires input from two users, each of whom is granted the CHANGE PASSWORD system privilege, to change the password of another user. **Values** ON, OFF **Default** OFF **Applies to** all users.
DEFAULT_LOGICAL_SERVER	If the connection string specifies no logical server, the user connects to the DEFAULT_LOGICAL_SERVER option specified in the user's login policy. **Values** • Name of an existing user-defined logical server • AUTO – value of the default logical server in the root login policy. • COORDINATOR – the current coordinator node. • NONE – denies access to any multiplex server. • OPEN – use alone or with the name of a user-defined logical server. Allows access to all multiplex nodes that are not members of any user-defined logical servers. • SERVER – allows access to all of the multiplex nodes, subject to the semantics of the SERVER logical server. **Default** AUTO **Applies to** all users. Requires MANAGE MULTIPLEX system privilege.
LOCKED	If set ON, users cannot establish new connections. This setting temporarily denies access to login policy users. Logical server overrides for this option are not allowed. **Values** ON, OFF **Default** OFF **Applies to** all users except those with the MANAGE ANY USER system privilege.
MAX_CONNECTIONS	The maximum number of concurrent connections allowed for a user. You can specify a per-logical-server setting for this option. **Values** 0 – 2147483647 **Default** UNLIMITED **Applies to** all users except those with the SERVER OPERATOR or DROP CONNECTION system privilege.
MAX_DAYS_SINCE_LOGIN	The maximum number of days that can elapse between two successive logins by the same user. **Values** 0 – 2147483647 **Default** UNLIMITED **Applies to** all users except those with the MANAGE ANY USER system privilege.
MAX_FAILED_LOGIN_ATTEMPTS	The maximum number of failed attempts, since the last successful attempt, to log in to the user account before the account is locked. **Values** 0 – 2147483647 **Default** UNLIMITED **Applies to** all users.

Option	Description
MAX_NON_DBA_CONNECTIONS	The maximum number of concurrent connections that a user without SERVER OPERATOR or DROP CONNECTION system privileges can make. This option is supported only in the root login policy. **Values** 0 – 2147483647 **Default** UNLIMITED **Applies to** all users except those with the SERVER OPERATOR or DROP CONNECTION privilege.
PAM_FAILOVER_TO_STD	Use standard authentication if PAM authentication is enabled but the PAM library is unavailable due to a system failure. Authentication failures returned by PAM do not fail over to standard authentication. **Values** ON, OFF **Default** ON **Applies to** all users.
PAM_SERVICE_NAME	The PAM service name to use when authenticating. The service name identifies the rule set to be used by PAM during validation. If empty (the default), do not use PAM. The database server continues to function when PAM support is unavailable. See *Enabling PAM User Authentication* in *Administration: User Management and Security*.
PASSWORD_EXPIRY_ON_NEXT_LOGIN	If set ON, the user's password expires at the next login. **Values** ON, OFF **Default** OFF **Applies to** all users. ℹ Note This functionality is not currently implemented when logging in to SAP IQ Cockpit. However, when logging in to SAP IQ outside of SAP IQ Cockpit (for example, using Interactive SQL), users are then prompted to enter a new password.
PASSWORD_GRACE_TIME	The number of days before password expiration during which login is allowed but the default post_login procedure issues warnings. **Values** 0 – 2147483647 **Default** 0 **Applies to** all users.
PASSWORD_LIFE_TIME	The maximum number of days before a password must be changed. **Values** 0 – 2147483647 **Default** UNLIMITED **Applies to** all users.

Option	Description
ROOT_AUTO_UNLOCK_TIME	The time period after which locked accounts that are granted the MANAGE ANY USER system privilege are automatically unlocked. You can define this option only in the root login policy. **Values** 0 – UNLIMITED **Default** 15 **Applies to** all users who are granted the MANAGE ANY USER system privilege.

4.1.6.2 LDAP Login Policy Options

Available login policy options for LDAP user authentication

Option	Description
LDAP_PRIMARY_SERVER	Specifies the name of the primary LDAP server. **Values** n/a **Default** None **Applies to** All users.
LDAP_SECONDARY_SERVER	Specifies the name of the secondary LDAP server. **Values** n/a **Default** None **Applies to** All users.
LDAP_AUTO_FAILBACK_PERIOD	Specifies the time period, in minutes, after which automatic failback to the primary server is attempted. **Values** 0 - 2147483647 **Default** 15 minutes **Applies to** All users.
LDAP_FAILOVER_TO_STD	Permits authentication with standard authentication when authentication with the LDAP server fails due to system resources, network outage, connection timeouts, or similar system failures. However, it does not permit an actual authentication failure returned from an LDAP server to fail over to standard authentication. **Values** ON, OFF **Default** ON **Applies to** All users.

Option	Description
LDAP_REFRESH_DN	Updates the ldap_refresh_dn value in the ISYSLOGINPOLICYOPTION system table with the current time, stored in Coordinated Universal Time (UTC). Each time a user authenticates with LDAP, if the value of ldap_refresh_dn in ISYSLOGINPOLICYOPTION is more recent than the value of user_dn in ISYSUSER, a search for a new user DN occurs. The user_dn value is then updated with the new user DN and the user_dn_changed_at value is again updated to the current time. **Values** NOW **Initial value for ROOT policy** NULL **Initial value for user-defined login policy** Current time stored in UTC **Applies to** All users.

4.1.6.3 Multiplex Login Policy Configuration

Configure login policies for multiplex servers.

Example

This example overrides the login policy settings on a logical server, increasing the maximum number of connections on logical server ls1:

```
ALTER LOGIN POLICY lp1 max_connections=20 LOGICAL SERVER ls1;
```

Usage

Applies only to multiplex.

Any login management commands you execute on any multiplex server automatically propagate to all servers in the multiplex. For best performance, execute these commands, or any DDL, on the coordinator.

An override at the logical server level override means that a particular login policy option has different settings for different logical servers. SYS.ISYSIQLSLOGINPOLICYOPTION stores login policy option values for logical-server override. For each logical-server override of a login policy option, a corresponding row exists in ISYSIQLSLOGINPOLICYOPTION.

4.1.7 CREATE ROLE Statement

Creates a new role, extends an existing user to act as a role, or manages role administrators on a role.

Syntax

```
CREATE [ OR REPLACE ] ROLE { <role_name> | FOR USER <userID> }
   [ WITH ADMIN [ ONLY ] <admin_name [...,]>, [ SYS_MANAGE_ROLES_ROLE ]
```

Parameters

- **role_name** unless you are using the OR REPLACE clause, `<role_name>` cannot already exist in the database.
- **OR REPLACE** `<role_name>` must already exist in the database. If `<role_name>` does not already exist, a new user-defined role is created. All current administrators are replaced by those specified in the `<admin_name [..]>` clause as follows:
  - All existing role administrators granted the WITH ADMIN OPTION not included on the new role administrators list become members of the role with no administrative rights on the role.
  - All existing role administrators granted the WITH ADMIN ONLY OPTION not included on the new role administrators list are removed as members of the role.

  When using the OR REPLACE clause, if an existing role administrator is included on the new role administrators list he or she retains his or her original administrative rights if they are higher than the replacement rights. For example, User A is an existing role administrator originally granted WITH ADMIN rights on the role. New role administrators are granted WITH ADMIN ONLY rights. If User A is included on this list, User A retains the higher WITH ADMIN rights.
- **FOR USER** when using the FOR USER clause without the OR REPLACE, `<userID>` must be the name of an existing user that currently does not have the ability to act as a role.
- **admin_name** list of users to be designated administrators of the role.
- **WITH ADMIN** each `<admin_name>` specified is granted administrative privileges over the role in addition to all underlying system privileges. WITH ADMIN clause is not valid when SYS_MANAGE_ROLES_ROLE is included on the list.
- **WITH ADMIN ONLY** each `<admin_name>` specified is granted administrative privileges only over the role, not the underlying system privileges.
- **SYS_MANAGE_ROLES_ROLE** allows global role administrators to administer the role. Can be specified in conjunction with the WITH ADMIN ONLY clause.

Examples

Example 1

creates the role Sales. Only global role administrator can administer the role.

```
CREATE ROLE Sales
```

Example 2

extends the existing user Jane to act as a role.

```
CREATE OR REPLACE ROLE FOR USER Jane
```

Example 3

creates the role Finance with Mary and Jeff as role administrators with administrative rights to the role. Global role administrators cannot administer this role.

```
CREATE ROLE Finance
WITH ADMIN Mary, Jeff
```

Example 3

creates the role Marketing with Mary and Jeff as role administrators. Global role administrators can also manage this role.

```
CREATE ROLE Finance
WITH ADMIN ONLY Mary, Jeff, SYS_MANAGE_ROLES_ROLE
```

Example 4

Finance is an existing role with Harry and Susan as role administrators with administrative rights. You want to keep Susan as an administrator, replace Harry, and add the global role administrator. The new role administrators will have administrative rights only.

This statement keeps Susan as an administrator, but Susan retains administrative rights to the role since the original administrative rights granted were higher. Harry is replaced by Bob and Sarah, with administrative rights only, and the global role administrator is added to the role. Harry remains a member of the role, but has no administrative rights.

```
CREATE OR REPLACE ROLE Finance
WITH ADMIN ONLY Susan, Bob, Sarah, SYS_MANAGE_ROLE_ROLE
```

Usage

If you specify role administrators (<admin_name>), but do not include the global role administrator (SYS_MANAGE_ROLES_ROLE), global role administrators will be unable to manage the new role. Therefore, it is recommended that you not specify role administrators during the creation process. Use the OR REPLACE clause to add them afterwards.

If you do not specify an ADMIN clause, the default WITH ADMIN ONLY clause is used and the default administrator is the global roles administrator (SYS_MANAGE_ROLES_ROLE).

When replacing role administrators, if the role has a global role administrator, it must be included on the new role administrators list or it is removed from the role.

However, when using the WITH ADMIN clause to grant role administrators, since the clause is not valid for global role administrators, you must use the `GRANT ROLE` statement to re-add the global role administrator (SYS_MANAGE_RILES_ROLE) to the role. Failure to perform this grant means global role administrators are unable to manage the role.

Standards

ANSI SQL–Compliance level: Transact-SQL extension.

Permissions

- Create a new role – Requires the MANAGE ROLES system privilege.
- OR REPLACE clause – Requires the MANAGE ROLES system privilege along with administrative rights over the role being replaced.

Related Information

Converting an Existing User to a User-Extended Role [page 13]

4.1.8 CREATE USER Statement

Creates a user.

Syntax

```
CREATE USER < user-name>  [ IDENTIFIED BY <password> ]
   [ LOGIN POLICY <policy-name> ]
   [ FORCE PASSWORD CHANGE { ON | OFF } ]
```

Parameters

- **user-name** Name of the user.

- **IDENTIFIED BY** The password for the user.
- **policy-name** Name of the login policy to assign the user. No change is made if you do not specify a login policy.
- **FORCE PASSWORD CHANGE** Controls whether the user must specify a new password upon logging in. This setting overrides the PASSWORD_EXPIRY_ON_NEXT_LOGIN option setting in the user's login policy.

  > i Note
  >
  > This functionality is not currently implemented when logging in to SAP IQ Cockpit. However, when logging in to SAP IQ outside of SAP IQ Cockpit (for example, using Interactive SQL), users are then prompted to enter a new password.

- **password** You do not have to specify a password for the user. A user without a password cannot connect to the database. This is useful if you are creating a role and do not want anyone to connect to the database using the role user ID. A user ID must be a valid identifier. User IDs and passwords cannot:
  - Begin with white space, single quotes, or double quotes
  - End with white space
  - Contain semicolons

  A password can be either a valid identifier, or a string (maximum 255 characters) placed in single quotes. Passwords are case-sensitive. The password should be composed of 7-bit ASCII characters, as other characters may not work correctly if the database server cannot convert them from the client's character set to UTF-8.

  You can use the VERIFY_PASSWORD_FUNCTION option to specify a function to implement password rules (for example, passwords must include at least one digit). If you do use a password verification function, you cannot specify more than one user ID and password in the `GRANT CONNECT` statement.

  The encryption algorithm used for hashing the user passwords is FIPS-certified encryption support:
  - The DLL is called dbfips10.dll.
  - The HASH function accepts the algorithms: SHA1_FIPS SHA256_FIPS.
  - If the -fips server option is specified and an algorithm that is not FIPS-certified is given to the HASH function, the database server uses SHA1_FIPS instead of SHA1, SHA256_FIPS instead of SHA256, and returns an error if MD5 is used (MD5 is not a FIPS-certified algorithm).
  - If the -fips option is specified, the database server uses SHA256_FIPS for password hashing.

Examples

Example 1

creates a user named `SQLTester` with the password welcome. The `SQLTester` user is assigned to the `Test1` login policy and the password expires on the next login:

```
CREATE USER SQLTester IDENTIFIED BY welcome
LOGIN POLICY Test1
FORCE PASSWORD CHANGE ON;
```

Standards

- SQL–Vendor extension to ISO/ANSI SQL grammar.
- SAP Database Products–Not supported by SAP ASE.

Permissions

Requires the MANAGE ANY USER system privilege.

4.1.9 DROP LDAP SERVER Statement

Removes the named LDAP server configuration object from the SYSLDAPSERVER system view after verifying that the LDAP server configuration object is not in a READY or ACTIVE state.

Syntax

```
DROP LDAP SERVER <ldapua-server-name>
   [ WITH DROP ALL REFERENCES ] [ WITH SUSPEND ]
```

Parameters

- **WITH DROP ALL REFERENCES** allows the removal of an LDAP server configuration object from service that has a reference in a login policy.
- **WITH SUSPEND** allows an LDAP server configuration object to be dropped even if in a READY or ACTIVE state.

Examples

Example 1

assuming that references to the LDAP server configuration object have been removed from all login policies, the following two sets of commands are equivalent. Using the WITH DROP ALL REFERENCES and WITH SUSPEND parameters eliminates the need to execute an ALTER LDAP SERVER statement before the DROP LDAP SERVER statement:

```
DROP LDAP SERVER ldapserver1 WITH DROP ALL REFERENCES WITH SUSPEND
```

is equivalent to

```
ALTER LDAP SERVER ldapserver1 WITH SUSPEND DROP LDAP SERVER ldapserver1 WITH
DROP ALL REFERENCES
```

Usage

The `DROP LDAP SERVER` statement fails when it is issued against an LDAP server configuration object that is in a READY or ACTIVE state. This ensures that an LDAP server configuration object in active use cannot be accidentally dropped. The `DROP LDAP SERVER` statement also fails if a login policy exists with a reference to the LDAP server configuration object.

Standards

ANSI SQL–Compliance level: Transact-SQL extension.

Permissions

Requires the MANAGE ANY LDAP SERVER system privilege.

4.1.10 DROP LOGIN POLICY Statement

Removes a login policy from the database.

Syntax

```
DROP LOGIN POLICY <policy-name>
```

Examples

Example 1

create and then delete the `Test11` login policy:

```
CREATE LOGIN POLICY Test11;
```

```
DROP LOGIN POLICY Test11 ;
```

Usage

A DROP LOGIN POLICY statement fails if you attempt to drop a policy that is assigned to a user. You can use either the ALTER USER statement to change the policy assignment of the user or DROP USER to drop the user.

Permissions

Requires the MANAGE ANY LOGIN POLICY system privilege.

4.1.11 DROP ROLE Statement

Removes a user-defined role from the database or converts a user-extended role to a regular user.

Syntax

```
DROP ROLE [ FROM USER ] <role_name>
   [ WITH REVOKE ]
```

Parameters

- **role_name** must be the name of a role that already exists in the database.
- **FROM USER** required to convert a user-extended role back to act as a regular user rather than remove it from the database. The <role_name> must exist in the database. The user retains any login privileges, system privileges, and roles granted to the user-extended role and becomes the owner of any objects owned by the user-extended role. Any users granted to the user-extended are immediately revoked.
- **WITH REVOKE** required when dropping a standalone or user-extended role to which users have been granted the underlying system privileges of the role. The grant can have been made with either the WITH ADMIN OPTION or WITH NO ADMIN OPTION clause.

Examples

Example 1

converts a user-extended role named `Joe` that has not been granted to other users or roles back to a regular user:

```
DROP ROLE FROM USER Joe
```

Example 2

drops a user-extended role named `Jack` that has not been granted to other users or roles from the database:

```
DROP ROLE Jack
```

Example 3

converts a user-extended role named `Sam` that has been granted to other user or roles back to a regular role:

```
DROP ROLE FROM USER Sam
WITH REVOKE
```

Example 4

drops a standalone role named `Sales2` that has been granted to other users or roles from the database:

```
DROP ROLE Sales2
WITH REVOKE
```

Usage

A user-defined role can be dropped from the database or converted back to a regular user at any time as long as all dependent roles left meet the minimum required number of administrative users with active passwords.

Standards

ANSI SQL–Compliance level: Transact-SQL extension.

Permissions

- Requires administrative rights over the role being dropped.
- If the role being dropped owns objects, none are in use by any user in any session at the time the DROP statement is executed.

4.1.12 DROP USER Statement

Removes a user.

Syntax

```
DROP USER <user-name>
```

Parameters

- **user-name** name of the user to remove.

Examples

Example 1

drops the user `SQLTester` from the database:

```
DROP USER SQLTester
```

Standards

- SQL–ISO/ANSI SQL compliant.
- SAP Database Products–Not supported by SAP ASE.

Permissions

Requires the MANAGE ANY USER system privilege.

> i Note
>
> When dropping a user, any objects owned by this user and any permissions granted by this user will be removed.

4.1.13 GRANT CHANGE PASSWORD Statement

Allows users to manage passwords for other users and administer the CHANGE PASSWORD system privilege.

Syntax

```
GRANT CHANGE PASSWORD ( <target_user_list> | ANY | ANY WITH ROLES<
target_role_list>  )
   TO <userID> [,…]
   [ WITH ADMIN [ONLY] OPTION | WITH NO ADMIN OPTION]
```

Parameters

- **target_user_list** Users the grantee has the potential to impersonate. The list must consist of existing users or user-extended roles with login passwords. Separate the userIDs in the list with commas.
- **ANY** All database users with login passwords become potential target users to manage passwords for each grantee.
- **ANY WITH ROLES target_role_list** List of target roles for each grantee. Any users who are granted any of the target roles become potential target users for each grantee. The `<target_role_list>` must consist of existing roles and the users who are granted said roles must consist of database users with login passwords. Use commas to separate multiple userIDs.
- **userID** Must be the name of an existing user or role that has a login password. Separate multiple userIDs with commas.
- **WITH ADMIN OPTION** (Valid with the ANY clause only) The user can both manage passwords and grant the CHANGE PASSWORD system privilege to another user.
- **WITH ADMIN ONLY OPTION** (Valid with the ANY clause only) The user can grant the CHANGE PASSWORD system privilege to another user, but cannot manage passwords of other users.
- **WITH NO ADMIN OPTION** The user can manage passwords, but cannot grant the CHANGE PASSWORD system privilege to another user.

Examples

Example 1

grants `Sally` and `Laurel` the ability to mange the password of `Bob`, `Sam`, and `Peter`:

```
GRANT CHANGE PASSWORD (Bob, Sam, Peter) TO (Sally, Laurel)
```

Example 2

grants `Mary` the right to grant the CHANGE PASSWORD system privilege to any user in the database. However, since the system privilege is granted with the WITH ADMIN ONLY OPTION clause, `Mary` cannot manage the password of any other user.

```
GRANT CHANGE PASSWORD (ANY) TO Mary WITH ADMIN ONLY OPTION
```

Example 3

grants `Steve` and `Joe` the ability to manage the password of any member of `Role1` or `Role2`:

```
GRANT CHANGE PASSWORD (ANY WITH ROLES Role1, Role2) TO Steve, Joe
```

Usage

A user can be granted the ability to mange the password of any user in the database (ANY) or only specific users (`<target_users_list>`) or members of specific roles (ANY WITH ROLES `<target_roles_list>`). Administrative rights to the CHANGE PASSWORD system privilege can only be granted when using the ANY clause.

If no clause is specified, ANY is used by default. If no administrative clause is specified in the grant statement, the WITH NO ADMIN OPTION clause is used.

By default, the CHANGE PASSWORD system privilege is granted to the SYS_AUTH_SA_ROLE compatibility role with the WITH NO ADMIN OPTION clause and to the SYS_AUTH_SSO_ROLE compatibility role with the ADMIN ONLY OPTION clause, if they exist.

Standards

ANSI SQL–Compliance level: Transact-SQL extension.

Permissions

- The CHANGE PASSWORD system privilege granted with administrative rights.
- Each target user specified (target_users_list) is an existing user or user-extended role with a login password.
- Each target role specified (target_roles_list) must be an existing user-extended or user-defined role.

4.1.14 GRANT CONNECT Statement

Grants CONNECT privilege to a user.

Syntax

```
GRANT CONNECT
   TO <userID>[, …]
   IDENTIFIED BY <password>[, …]
```

Parameters

- **userID** Must be the name of an existing user or role that has a login password. Separate multiple userIDs with commas.

Examples

Example 1

creates two new users for the database named `Laurel` and `Hardy`:

```
GRANT CONNECT TO Laurel, Hardy
IDENTIFIED BY Stan, Ollie
```

Example 2

creates user `Jane` with no password:

```
GRANT CONNECT TO Jane
```

Example 3

changes the password for `Bob` to `newpassword`:

```
GRANT CONNECT TO Bob IDENTIFIED BY <newpassword>
```

Usage

`GRANT CONNECT` can be used to create a new user or also be used by any user to change their own password.

> → Tip
>
> Use the `CREATE USER` statement rather than the `GRANT CONNECT` statement to create users.

> If you inadvertently enter the user ID of an existing user when you are trying to add a new user, you are actually changing the password of the existing user. You do not receive a warning because this behavior is considered normal.

The stored procedures `sp_addlogin` and `sp_adduser` can also be used to add users. These procedures display an error if you try to add an existing user ID.

> i Note
>
> Use system procedures, not `GRANT` and `REVOKE` statements to add and remove user IDs.

A user without a password cannot connect to the database. This is useful when you are creating groups and you do not want anyone to connect to the role user ID. To create a user without a password, do not include the IDENTIFIED BY clause.

When specifying a password, it must be a valid identifier Passwords have a maximum length of 255 bytes. If the `VERIFY_PASSWORD_FUNCTION` database option is set to a value other than the empty string, the `GRANT CONNECT TO` statement calls the function identified by the option value. The function returns NULL to indicate that the password conforms to rules. If the `VERIFY_PASSWORD_FUNCTION` option is set, you can specify only one `<userid>` and `<password>` with the `GRANT CONNECT` statement.

Invalid names for database user IDs and passwords include those that:

- Begin with white space or single or double quotes
- End with white space
- Contain semicolons

Standards

- SQL–Other syntaxes are vendor extensions to ISO/ANSI SQL grammar.
- SAP Database Products–The security model is different in SAP ASE and SAP IQ, so other syntaxes differ.

Permissions

- If you are creating a new user, you must have the MANAGE ANY USER system privilege.
- Any user can change his or her own password.
- If you are changing another user's password, you must have the CHANGE PASSWORD system privilege.

> i Note
>
> If you are changing another user's password, the other user cannot be connected to the database.

4.1.15 GRANT CREATE Statement

Grants CREATE privilege on a specified dbspace to the specified users and roles.

Syntax

```
GRANT CREATE
  ON <dbspace_name>
  TO <userID [,...]>
```

Parameters

- **userID** Must be the name of an existing user or role that has a login password. Separate multiple userIDs with commas.

Examples

Example 1

grants users `Lawrence` and `Swift` `CREATE` privilege on dbspace `<DspHist>`:

```
GRANT CREATE ON DspHist
TO LAWRENCE, SWIFT
```

Example 2

grants `CREATE` privilege on dbspace `DspHist` to users `Fiona` and `Ciaran`:

```
GRANT CREATE ON DspHist TO Fiona, Ciaran
```

Standards

- SQL–other syntaxes are vendor extensions to ISO/ANSI SQL grammar.
- SAP Database Products–the security model is different in SAP ASE and SAP IQ, so other syntaxes differ.

Permissions

Requires the MANAGE ANY DBSPACE system privilege.

4.1.16 GRANT EXECUTE Statement

Grants EXECUTE privilege on a procedure or user-defined function.

Syntax

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

Parameters

- **userID** Must be the name of an existing user or role that has a login password. Separate multiple userIDs with commas.

Standards

- SQL–syntax is a Persistent Stored Module feature.
- SAP Database Products–the security model is different in SAP ASE and SAP IQ, so other syntaxes differ.

Permissions

Requires one of:

- MANAGE ANY OBJECT PRIVILEGE system privilege.
- You own the procedure.

4.1.17 GRANT Object-Level Privilege Statement

Grants database object-level privileges on individual tables or views to a user or role.

Syntax

```
GRANT object-level-privilege [, …]
   ON [ <owner>.]<object-name>
   TO <userID>[, …]
   [ WITH GRANT OPTION ]
object-level-privilege
   ALL [ PRIVILEGES ]
   | ALTER
   | DELETE
   | INSERT
   | LOAD
   | REFERENCE [ ( <column-name> [, …] ) ]
   | SELECT [ ( <column-name> [, …] ) ]
   | TRUNCATE
   | UPDATE [ ( <column-name>, …) ] }
```

Parameters

- **userID** Must be the name of an existing user or immutable role. The list must consist of existing users with login passwords. Separate the userIDs in the list with commas.
- **ALL** Grants all privileges to users
- **ALTER** Users can alter this table with the `ALTER TABLE` statement. This privilege is not allowed for views.
- **DELETE** Users can delete rows from this table or view.
- **INSERT** Users can insert rows into the named table or view.
- **LOAD** Users can load data into the named table or view.
- **REFERENCES** Users can create indexes on the named tables, and foreign keys that reference the named tables. If column names are specified, then users can reference only those columns. REFERENCES privileges on columns cannot be granted for views, only for tables.
- **SELECT** Users can look at information in this view or table. If column names are specified, then the users can look at only those columns. SELECT permissions on columns cannot be granted for views, only for tables.
- **TRUNCATE** Users can truncate the named table or view.
- **UPDATE** Users can update rows in this view or table. If column names are specified, users can update only those columns. UPDATE privileges on columns cannot be granted for views, only for tables. To update a table, users must have both SELECT and UPDATE privilege on the table.
- **WITH GRANT OPTION** The named user ID is also given privileges to grant the same privileges to other user IDs.

Usage

You can list the table privileges, or specify ALL to grant all privileges at once.

Standards

- SQL–Syntax is an entry-level feature.
- SAP Database Products–Syntax is supported in SAP ASE.

Permissions

Requires one of:

- MANAGE ANY OBJECT PRIVILEGE system privilege
- You have been granted the specific object privilege with the WITH GRANT OPTION clause on the table.
- You own of the table.

4.1.18 GRANT ROLE Statement

Grants roles to users or other roles, with or without administrative rights.

Syntax

```
GRANT ROLE role_name [, …]
   TO <grantee> [, …]
   [ {WITH NO ADMIN | WITH ADMIN [ ONLY ] } OPTION ]
   [ WITH NO SYSTEM PRIVILEGE INHERITANCE ]
role_name
   dbo†††
   | diagnostics†††
   | PUBLIC†††
   | rs_systabgroup†††
   | SA_DEBUG†††
   | SYS†††
   | SYS_AUTH_SA_ROLE
   | SYS_AUTH_SSO_ROLE
   | SYS_AUTH_DBA_ROLE††
   | SYS_AUTH_RESOURCE_ROLE†
   | SYS_AUTH_BACKUP_ROLE†
   | SYS_AUTH_VALIDATE_ROLE†
   | SYS_AUTH_WRITEFILE_ROLE
   | SYS_AUTH_WRITEFILECLIENT_ROLE
   | SYS_AUTH_READFILE_ROLE
   | SYS_AUTH_READFILECLIENT_ROLE
   | SYS_AUTH_PROFILE_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
   | SYS_REPLICATE_ADMIN_ROLE†††
   | SYS_RUN_REPLICATE_ROLE†††
   | SYS_SPATIAL_ADMIN_ROLE†††
   | <user-defined role name>
```

Parameters

- **role_name** Must already exist in the database. Separate multiple role names with commas.

- **grantee** Must be the name of an existing user or role that has a login password. Separate multiple userIDs with commas.
- **WITH NO ADMIN OPTION** Each `<grantee>` is granted the underlying system privileges of each `<role_name>`, but cannot grant `<role_name>` to another user.
- **WITH ADMIN ONLY OPTION** Each `<userID>` is granted administrative privileges over each `<role_name>`, but not the underlying system privileges of `<role_name>`.
- **WITH ADMIN OPTION** Each userID is granted the underlying system privileges of each `<role_name>`, along with the ability to grant `<role_name>` to another user.
- **WITH NO SYSTEM PRIVILEGE INHERITANCE** The underlying system privileges of the granting role are not inherited by the members of the receiving role. However, if the receiving role is a user-extended role, the underlying system privileges are granted to the extended user.

Examples

Example 1

grants `Sales_Role` to `Sally`, with administrative privileges, which means she can grant or revoke `Sales_Role` to other users as well as perform any authorized tasks granted by the role:

```
GRANT ROLE Sales_Role TO Sally WITH ADMIN OPTION
```

Example 2

grants the compatibility role SYS_AUTH_PROFILE_ROLE to the role `Sales_Admin` with no administrative rights. `Sales_Admin` is a standalone role and `Mary` and `Peter` have been granted `Sales_Admin`. Since SYS_AUTH_PROFILE_ROLE is an inheritable compatibility role, `Mary` and `Peter` are granted the underlying system privileges of `Sales_Role`. Since the role is granted with no administrative rights, they cannot grant or revoke the role.

```
GRANT ROLE SYS_AUTH_PROFILE_ROLE TO Sales_Role WITH NO ADMIN OPTION
```

Example 3

grants the compatibility role SYS_AUTH_BACKUP_ROLE to `Tom` with no administrative rights. `Tom` is a user-extended role to which `Betty` and Laurel have been granted. Since `SYS_AUTH_BACKUP_ROLE` is a non-inheritable compatibility role, the underlying system privileges of the role are not granted to Betty and Laurel. However, since `Tom` is an extended user, the underlying system privileges are granted directly to `Tom`.

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

Usage

- The WITH NO SYSTEM PRIVILEGE INHERITANCE clause can be used when granting select compatibility roles to other roles. It prevents automatic inheritance of the compatibility role's underlying system privileges by members of the role. When granted to user-extended roles, the WITH NO SYSTEM PRIVILEGE

INHERITANCE clause applies to members of the role only. The user acting as a role automatically inherits the underlying system privileges regardless of the clause.

- The WITH NO ADMIN OPTION WITH NO SYSTEM PRIVILEGE INHERITANCE and WITH NO SYSTEM PRIVILEGE INHERITANCE clauses are semantically equivalent.
- †The WITH ADMIN OPTION or WITH ADMIN ONLY clauses can not be specified in combination with the WITH NO SYSTEM PRIVILEGE INHERITANCE clause when granting the SYS_AUTH_BACKUP_ROLE, SYS_AUTH_RESOURCE_ROLE, or SYS_AUTH_VALIDATE_ROLE roles.
- ††The WITH ADMIN OPTION clause can only be specified in combination with the WITH NO SYSTEM PRIVILEGE INHERITANCE clause when granting the SYS_AUTH_DBA_ROLE or SYS_RUN_REPLICATION_ROLE roles.
- †††The WITH ADMIN OPTION and WITH ADMIN ONLY OPTION clauses are not supported for system roles.

Use of the WITH ADMIN OPTION or WITH ADMIN ONLY OPTION clause allows the grantee to grant or revoke the role, but does not allow the grantee to drop the role.

By default, if no administrative clause is specified in the grant statement, each compatibility role is granted with these default administrative rights:

WITH ADMIN OPTION	WITH ADMIN ONLY OPTION	WITH NO ADMIN OPTION
SYS_AUTH_SA_ROLE SYS_AUTH_SSO_ROLE	SYS_AUTH_DBA_ROLE	SYS_AUTH_RESOURCE_ROLE SYS_AUTH_BACKUP_ROLE SYS_AUTH_VALIDATE_ROLE SYS_AUTH_WRITEFILE_ROLE SYS_AUTH_WRITEFILECLIENT_ROLE SYS_AUTH_READFILE_ROLE SYS_AUTH_READFILECLIENT_ROLE SYS_AUTH_PROFILE_ROLE SYS_AUTH_USER_ADMIN_ROLE SYS_AUTH_SPACE_ADMIN_ROLE SYS_AUTH_MULTIPLEX_ADMIN_ROLE SYS_AUTH_OPERATOR_ROLE SA_DEBUG SYS_RUN_REPLICATION_ROLE

The SYS_AUTH_PERMS_ADMIN_ROLE role grants these underlying roles with these default administrative rights:

WITH ADMIN OPTION	WITH NO ADMIN OPTION
SYS_AUTH_BACKUP_ROLE	MANAGE ROLES
SYS_AUTH_OPERATOR_ROLE	MANAGE ANY OBJECT PRIVILEGE
SYS_AUTH_USER_ADMIN_ROLE	CHANGE PASSWORD
SYS_AUTH_SPACE_ADMIN_ROLE	
SYS_AUTH_MULTIPLEX_ADMIN_ROLE	
SYS_AUTH_RESOURCE_ROLE	
SYS_AUTH_VALIDATE_ROLE	
SYS_AUTH_PROFILE_ROLE	
SYS_AUTH_WRITEFILE_ROLE	
SYS_AUTH_WRITEFILECLIENT_ROLE	
SYS_AUTH_READFILE_ROLE	
SYS_AUTH_READFILECLIENT_ROLE	

Standards

- SQL–Other syntaxes are vendor extensions to ISO/ANSI SQL grammar.
- SAP Database Products–Syntax is supported in SAP ASE.

Permissions

- Requires MANAGE ROLES system privilege to grant these system roles:
  - dbo
  - diagnostics
  - PUBLIC
  - rs_systabgroup
  - SA_DEBUG SYS
  - SYS
  - SYS_REPLICATION_ADMIN_ROLE
  - SYS_RUN_REPLICATION_ROLE
  - SYS_SPATIAL_ADMIN_ROLE
- Requires administrative privilege over the role to grant these roles:
  - SYS_AUTH_SA_ROLE
  - SYS_AUTH_SSO_ROLE
  - SYS_AUTH_DBA_ROLE
  - SYS_AUTH_RESOURCE_ROLE

  - SYS_AUTH_BACKUP_ROLE
  - SYS_AUTH_VALIDATE_ROLE
  - SYS_AUTH_WRITEFILE_ROLE
  - SYS_AUTH_WRITEFILECLIENT_ROLE
  - SYS_AUTH_READFILE_ROLE
  - SYS_AUTH_READFILECLIENT_ROLE
  - SYS_AUTH_PROFILE_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
  - <user-defined role name>

Related Information

Adding the Global Role Administrator to an Existing Role [page 25]

4.1.19 GRANT SET USER Statement

Grants the ability for one user to impersonate another user and to administer the SET USER system privilege.

Syntax

```
GRANT SET USER ( <target_users_list>
       | ANY
       | ANY WITH ROLES <target_roles_list> )
  TO <userID> [,…]
  [ WITH ADMIN [ ONLY ] OPTION | WITH NO ADMIN OPTION ]
```

Parameters

- **target_users_list** Must consist of existing users with login passwords and is the potential list of target users who can no longer be impersonated by grantee users. Separate the user IDs in the list with commas.
- **ANY** The potential list of target users for each grantee consists of all database users with login passwords.
- **ANY WITH ROLES target_roles_list** The `<target_role_list>` must consist of existing roles, and the potential list of target users for each grantee must consist of database users with login passwords that have a subset of roles in `<target_role_list>`. Separate the list of roles with commas.

- **userID** Each `<userID>` must be the name of an existing user or immutable role. The list must consist of existing users with login passwords. Separate the userIDs in the list with commas.
- **WITH ADMIN OPTION** (Valid in conjunction with the ANY clause only) The user can both issue the SETUSER command to impersonate another user and grant the SET USER system privilege to another user.
- **WITH ADMIN ONLY OPTION** (Valid in conjunction with the ANY clause only) The user can grant the SET USER system privilege to another user, but cannot issue the SETUSER command to impersonate another user.
- **WITH NO ADMIN OPTION** The user can issue the SETUSER command to impersonate another user, but cannot grant the SET USER system privilege to another user.

Examples

Example 1

grants `Sally` and `Laurel` the ability to impersonate `Bob`, `Sam`, and `Peter`:

```
GRANT SET USER (Bob, Sam, Peter) TO (Sally, Laurel)
```

Example 2

grants `Mary` the right to grant the SET USER system privilege to any user in the database. However, since the system privilege is granted with the WITH ADMIN ONLY OPTION clause, `Mary` cannot impersonate any other user.

```
GRANT SET USER (ANY) TO Mary WITH ADMIN ONLY OPTION
```

Example 3

grants `Steve` and `Joe` the ability to impersonate any member of `Role1` or `Role2`:

```
GRANT SET USER (ANY WITH ROLES Role1, Role2) TO Steve, Joe
```

Usage

A user can be granted the ability to impersonate any user in the database (ANY) or only specific users (`<target_users_list>`) or members of specific roles (ANY WITH ROLES `<target_roles_list>`). Administrative rights to the SET USER system privilege can only be granted when using the ANY clause.

If no clause is specified, ANY is used by default. If no administrative clause is specified in the grant statement, the WITH NO ADMIN OPTION clause is used.

If regranting the SET USER system privilege to a user, the effect of the regrant is cumulative.

By default, the SET USER system privilege is granted to the SYS_AUTH_SSO_ROLE compatibility role with the WITH NO ADMIN OPTION clause, if they exist.

The granting of the SET USER system privilege to a user only grants the potential to impersonate another user. Validation of the at-least criteria required to successfully impersonate another user does no occur until the `SETUSER` statement is issued.

Standards

ANSI SQL–Compliance level: Transact-SQL extension.

Permissions

- The SET USER system privilege granted with administrative rights.
- Each target user specified (target_users_list) is an existing user or user-extended role with a login password.
- Each target role specified (target_roles_list) must be an existing user-extended or user-defined role.

4.1.20 GRANT System Privilege Statement

Grants specific system privileges to users or roles, with or without administrative rights.

Syntax

```
GRANT <system_privilege_name>  [, …]
   TO <userID> [, …]
   [ { WITH NO ADMIN | WITH ADMIN [ ONLY ] } OPTION ]
```

Parameters

- **system_privilege_name** must be the name of an existing system privilege.
- **userID** must be the name of an existing user or immutable role. The list must consist of existing users with login passwords. Separate multiple userIDs with commas.
- **WITH NO ADMIN OPTION** the user can manage the system privilege, but cannot grant the system privilege to another user.
- **WITH ADMIN ONLY OPTION** if the WITH ADMIN ONLY OPTION clause is used, each `<userID>` is granted administrative privileges over each `<system_privilege>`, but not the `<system_privilege>` itself.
- **WITH ADMIN OPTION** each `<userID>` is granted administrative privileges over each `<system_privilege>` in addition to all underlying system privileges of `<system_privilege>`.

Examples

Example 1

grants the DROP CONNECTION system privilege to `Joe` with administrative privileges:

```
GRANT DROP CONNECTION TO Joe WITH ADMIN OPTION
```

Example 2

grants the CHECKPOINT system privilege to `Sally` with no administrative privileges:

```
GRANT CHECKPOINT TO Sally WITH NO ADMIN OPTION
```

Example 3

grants the MONITOR system privilege to `Jane` with administrative privileges only:

```
GRANT MONITOR TO Jane WITH ADMIN ONLY OPTION
```

Usage

By default, if no administrative clause is specified in the grant statement, the WITH NO ADMIN OPTION clause is used.

Standards

- SQL–Other syntaxes are vendor extensions to ISO/ANSI SQL grammar.
- SAP Database Products–Syntax is supported in SAP ASE.

Permissions

Requires administrative privilege over the system privilege being granted.

In this section:

4.1.20.1 List of All System Privileges

A list of all system privileges.

System privileges control the rights of users to perform authorized database tasks.

The following is a list of available system privileges:

- 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

4.1.21 GRANT USAGE ON SEQUENCE Statement

Grants the USAGE system privilege on a specified sequence to a user or role.

Syntax

```
GRANT USAGE ON SEQUENCE <sequence-name>
   TO <userID [,...]>
```

Parameters

- **userID** Must be the name of an existing user or role that has a login password. Separate multiple userIDs with commas.

Standards

- SQL–syntax is a Persistent Stored Module feature.
- SAP Database Products–the security model is different in SAP ASE and SAP IQ, so other syntaxes differ.

Permissions

Requires one of:

- MANAGE ANY OBJECT PRIVILEGE system privilege.
- You own the sequence.

4.1.22 REVOKE CHANGE PASSWORD Statement

Removes the ability of a user to manage passwords and administer the system privilege.

Syntax

```
REVOKE [ ADMIN OPTION FOR ] CHANGE PASSWORD
   [(<target_user_list>
      | ANY
      | ANY WITH ROLES <target_role_list>  )]
    FROM <userID> [,...]
```

Parameters

- **target_user_list** Users the grantee has the potential to impersonate. The list must consist of existing users or user-extended roles with login passwords. Separate the userIDs in the list with commas.
- **ANY** All database users with login passwords become potential target users to manage passwords for each grantee.
- **ANY WITH ROLES target_role_list** List of target roles for each grantee. Any users who are granted any of the target roles become potential target users for each grantee. The `<target_role_list>` must consist of existing roles and the users who are granted said roles must consist of database users with login passwords. Use commas to separate multiple userIDs.
- **userID** Must be the name of an existing user or role that has a login password. Separate multiple userIDs with commas.

Examples

Example 1

removes the ability of `Joe` to manage the passwords of `Sally` or `Bob`:

```
REVOKE CHANGE PASSWORD (Sally, Bob) FROM Joe
```

Example 2

if the CHANGE PASSWORD system privilege was originally granted to `Sam` with the WITH ADMIN OPTION clause, this example removes the ability of `Sam` to grant the CHANGE PASSWORD system privilege to another user, but still allows `Sam` to manage passwords for those users specified in the original `GRANT CHANGE PASSWORD` statement. However, if the CHANGE PASSWORD system privilege was originally granted to `Sam` with the WITH ADMIN ONLY OPTION clause, this example removes all permissions to the system privilege from `Sam`.

```
REVOKE ADMIN OPTION FOR CHANGE PASSWORD FROM Sam
```

Usage

Depending on how the CHANGE PASSWORD system privilege was initially granted, using the ADMIN OPTION FOR clause when revoking the CHANGE PASSWORD system privilege has different results. If the CHANGE PASSWORD system privilege was originally granted with the WITH ADMIN OPTION clause, including the ADMIN OPTION FOR clause in the revoke statement revokes only the ability to administer the CHANGE PASSWORD system privilege (that is, grant the system privilege to another user). The ability to actually manage passwords for other users remains. However, if the CHANGE PASSWORD system privilege was originally granted with the WITH ADMIN ONLY OPTION clause, including the ADMIN OPTION FOR clause in the revoke statement is semantically equivalent to revoking the entire CHANGE PASSWORD system privilege. Finally, if the CHANGE PASSWORD system privilege was originally granted with the WITH NO ADMIN OPTION clause, and the ADMIN OPTION FOR clause is included in the revoke statement, nothing is revoked because there were no administrative rights granted in the first place.

You can revoke the CHANGE PASSWORD system privilege from any combination of users and roles granted.

Standards

ANSI SQL–Compliance level: Transact-SQL extension.

Permissions

The CHANGE PASSWORD system privilege granted with administrative rights.

4.1.23 REVOKE CONNECT Statement

Removes a user from the database.

Syntax

```
REVOKE CONNECT
   FROM <userID> [,...]
```

Parameters

- **userID** Must be the name of an existing user or role that has a login password. Separate multiple userIDs with commas.

Usage

Use system procedures or CREATE USER and DROP USER statements, not `GRANT` and `REVOKE` statements, to add and remove user IDs.

You cannot revoke the connect privileges from a user if he or she owns database objects, such as tables. Attempting to do so with a REVOKE statement, or sp_droplogin or sp_iqdroplogin stored procedure returns an error such as Cannot drop a user that owns tables in runtime system.

Standards

ANSI SQL–compliance level: Transact-SQL extension.

Permissions

Requires the MANAGE ANY USER system privilege.

> i Note
>
> If revoking CONNECT permissions or revoking table permissions from another user, the target user cannot be connected to the database.

4.1.24 REVOKE CREATE Statement

Removes CREATE privileges on the specified dbspace from the specified user IDs.

Syntax

```
REVOKE CREATE ON <dbspace-name>
   FROM <userID> [,...]
```

Parameters

- **userID** Must be the name of an existing user or role that has a login password. Separate multiple userIDs with commas.

Examples

Example 1

revokes the CREATE privilege on dbspace `DspHist` from user `Smith`:

```
REVOKE CREATE ON DspHist FROM Smith
```

Example 2

revokes the CREATE privilege on dbspace `DspHist` from user ID `fionat` from the database:

```
REVOKE CREATE ON DspHist FROM fionat
```

Standards

ANSI SQL–Compliance level: Transact-SQL extension.

Permissions

Requires the MANAGE ANY DBSPACE system privilege.

4.1.25 REVOKE EXECUTE Statement

Removes EXECUTE permissions that were given using the `GRANT` statement.

Syntax

```
REVOKE EXECUTE ON [ <owner>.]<procedure-name>
   FROM <userID> [,...]
```

Parameters

- **userID** Must be the name of an existing user or role that has a login password. Separate multiple userIDs with commas.

Standards

- SQL—Syntax is a Persistent Stored Module feature.
- SAP Database Products—Syntax is supported by SAP ASE. User management and security models are different for SAP ASE and SAP IQ.

Permissions

Requires one of:

- Own the procedure, or
- Have been granted the MANAGE ANY OBJECT PRIVILEGE system privilege.

4.1.26 REVOKE Object-Level Privilege Statement

Removes object-level privileges that were given using the GRANT statement.

Syntax

```
REVOKE { object-level-privilege [,...]
    [ <owner>.]<table-name>
   FROM <userID> [,...]
object-level-privilege
   ALL [ PRIVILEGES ]
   | ALTER
   | DELETE
   | INSERT
   | LOAD
   | REFERENCE [ ( <column-name> [, …] ) ]
   | SELECT [ ( <column-name> [, …] ) ]
   | TRUNCATE
   | UPDATE [ ( <column-name>, …) ] }
```

Parameters

- **userID** Must be the name of an existing user or immutable role. The list must consist of existing users with login passwords. Separate the userIDs in the list with commas.
- **ALL** Grants all privileges to users
- **ALTER** Users can alter this table with the ALTER TABLE statement. This privilege is not allowed for views.
- **DELETE** Users can delete rows from this table or view.
- **INSERT** Users can insert rows into the named table or view.

- **LOAD** Users can load data into the named table or view.
- **REFERENCES** Users can create indexes on the named tables, and foreign keys that reference the named tables. If column names are specified, then users can reference only those columns. REFERENCES privileges on columns cannot be granted for views, only for tables.
- **SELECT** Users can look at information in this view or table. If column names are specified, then the users can look at only those columns. SELECT permissions on columns cannot be granted for views, only for tables.
- **TRUNCATE** Users can truncate the named table or view.
- **UPDATE** Users can update rows in this view or table. If column names are specified, users can update only those columns. UPDATE privileges on columns cannot be granted for views, only for tables. To update a table, users must have both SELECT and UPDATE privilege on the table.

Examples

Example 1

prevents user `Dave` from inserting into the `Employees` table:

```
REVOKE INSERT ON Employees FROM Dave
```

Example 2

prevents user `Dave` from updating the `Employees` table:

```
REVOKE UPDATE ON Employees FROM Dave
```

Standards

- SQL–Syntax is an entry-level feature.
- SAP Database Products–Syntax is supported in SAP ASE.

Permissions

Requires one of:

- Own the table, or
- Have the MANAGE ANY OBJECT PRIVILEGE system privilege granted with the GRANT OPTION clause.

4.1.27 REVOKE ROLE Statement

Removes a users membership in a role or his or her ability to administer the role.

Syntax

```
REVOKE [ ADMIN OPTION FOR ] ROLE role_name [,...]
   FROM <grantee> [,...]
role_name
   dbo†††
    | diagnostics†††
    | PUBLIC†††
    | rs_systabgroup†††
    | SA_DEBUG†††
    | SYS†††
    | SYS_AUTH_SA_ROLE
    | SYS_AUTH_SSO_ROLE
    | SYS_AUTH_DBA_ROLE
    | SYS_AUTH_RESOURCE_ROLE
    | SYS_AUTH_BACKUP_ROLE
    | SYS_AUTH_VALIDATE_ROLE
    | SYS_AUTH_WRITEFILE_ROLE
    | SYS_AUTH_WRITEFILECLIENT_ROLE
    | SYS_AUTH_READFILE_ROLE
    | SYS_AUTH_READFILECLIENT_ROLE
    | SYS_AUTH_PROFILE_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
    | SYS_REPLICATE_ADMIN_ROLE†††
    | SYS_RUN_REPLICATE_ROLE†††
    | SYS_SPATIAL_ADMIN_ROLE†††
    | <user-defined role name>
```

†††The ADMIN OPTION FOR clause is not supported for system roles.

Parameters

- **role_name** Must already exist in the database. Separate multiple role names with commas.
- **userID** Must be the name of an existing user or role that has a login password. Separate multiple userIDs with commas.
- **ADMIN OPTION FOR** Each `<userID>` must have been granted administrative privilege over the specified `<role_name>`.

  > **Note**
  >
  > This clause revokes administrative privileges of the role only, not membership in the role, unless the role was originally granted with the WITH ADMIN ONLY OPTION clause. For roles granted with the WITH ADMIN ONLY OPTION clause, the ADMIN OPTION FOR clause is optional as it is semantically equivalent to revoking membership in a role in its entirety.

Examples

Example 1

revokes the user-defined (standalone) role `Role1` from `User1`:

```
REVOKE ROLE Role1 FROM User1
```

After you execute this command, `User1` no longer has the rights to perform any authorized tasks using any system privileges granted to `Role1`.

Example 2

revokes the ability for `User1` to administer the compatibility role SYS_AUTH_WRITEFILE_ROLE:

```
REVOKE ADMIN OPTION FOR ROLE SYS_AUTH_WRITEFILE_ROLE FROM User1
```

User1 retains the ability to perform any authorized tasks granted by SYS_AUTH_WRITEFILE_ROLE.

Standards

- SQ Other syntaxes are vendor extensions to ISO/ANSI SQL grammar.
- SAP Database Products–Syntax is supported in SAP ASE.

Permissions

Requires the MANAGE ROLES system privilege to revoke these roles:

- diagnostics
- dbo
- PUBLIC
- rs_systabgroup
- SA_DEBUG
- SYS
- SYS_RUN_REPLICATE_ROLE
- SYS_SPATIAL_ADMIN_ROLE

Requires administrative privilege over the role to revoke these roles:

- SYS_AUTH_SA_ROLE
- SYS_AUTH_SSO_ROLE
- SYS_AUTH_DBA_ROLE
- SYS_AUTH_RESOURCE_ROLE
- SYS_AUTH_BACKUP_ROLE
- SYS_AUTH_VALIDATE_ROLE
- SYS_AUTH_WRITEFILE_ROLE

- SYS_AUTH_WRITEFILECLIENT_ROLE
- SYS_AUTH_READFILE_ROLE
- SYS_AUTH_READFILECLIENT_ROLE
- SYS_AUTH_PROFILE_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
- <user-defined role name>

Related Information

Removing the Global Role Administrator from a Role [page 30]

4.1.28 REVOKE SET USER Statement

Removes the ability for one user to impersonate another user and to administer the SET USER system privilege.

Syntax

```
REVOKE [ ADMIN OPTION FOR ] SETUSER
   (<target_user_list>
      | ANY
      | ANY WITH ROLES <target_role_list> ] )
   FROM <userID> [,...]
```

Parameters

- **target_user_list** Must consist of existing users with login passwords and is the potential list of target users who can no longer be impersonated by grantee users. Separate the user IDs in the list with commas.
- **ANY** The potential list of target users for each grantee consists of all database users with login passwords.
- **ANY WITH ROLES target_role_list** The <target_role_list> must consist of existing roles, and the potential list of target users for each grantee must consist of database users with login passwords that have a subset of roles in <target_role_list>. Separate the list of roles with commas.
- **userID** Each <userID> must be the name of an existing user or immutable role. The list must consist of existing users with login passwords. Separate the userIDs in the list with commas.

Examples

Example 1

stops `Bob` from being able to impersonate `Sally` or `Bob`:

```
REVOKE SET USER (Sally, Bob) FROM Bob
```

Example 2

if the SET USER system privilege was originally granted to `Sam` with the WITH ADMIN OPTION clause, this example removes the ability of `Sam` to grant the SET USER system privilege to another user, but still allows `Sam` to impersonate those users already granted to him or her. However, if the SET USER system privilege was originally granted to `Sam` with the WITH ADMIN ONLY OPTION clause, this example removes all permissions to the system privilege from `Sam`.

```
REVOKE ADMIN OPTION FOR SET USER FROM Sam
```

Usage

Depending on how the SET USER system privilege was initially granted, using the ADMIN OPTION FOR clause when revoking the SET USER system privilege has different results. If you the SET USER system privilege was originally granted with the WITH ADMIN OPTION clause, including the ADMIN OPTION FOR clause in the revoke statement revokes only the ability to administer the SET USER system privilege (that is, grant the system privilege to another user). The ability to actually impersonate another user remains. However, if the SET USER system privilege was originally granted with the WITH ADMIN ONLY OPTION clause, including the ADMIN OPTION FOR clause in the revoke statement is semantically equivalent to revoking the entire SET USER system privilege. Finally, if the SET USER system privilege was originally grant with the WITH NO ADMIN OPTION clause, and the ADMIN OPTION FOR clause is included in the revoke statement, nothing is revoked because there were no administrative system privileges granted in the first place.

Standards

ANSI SQL–Compliance level: Transact-SQL extension.

Permissions

The SET USER system privilege granted with administrative rights.

4.1.29 REVOKE System Privilege Statement

Removes specific system privileges from specific users and the right to administer the privilege.

Syntax

```
REVOKE [ ADMIN OPTION FOR ] <system_privilege_name> [,...]
   FROM <userID> [,...]
```

Parameters

- **system_privilege_name** must be an existing system privilege.
- **userID** Must be the name of an existing user or role that has a login password. Separate multiple userIDs with commas.
- **ADMIN OPTION FOR** each `<system_privilege>` must currently be granted to each `<userID>` specified with administrative privileges.

> i Note
>
> This clause revokes only the administrative privileges of the system privilege; the system privilege itself remains granted. However, if the system privilege was originally granted with the WITH ADMIN ONLY OPTION clause, the ADMIN OPTION FOR clause completely revokes the system privilege. Under this scenario, use of the ADMIN OPTION FOR clause is not required to revoke administrative privileges.

Examples

Example 1

revokes the BACKUP DATABASE system privilege from user `Jim`:

```
REVOKE BACKUP DATABASE FROM Jim
```

Example 2

assuming the BACKUP DATABASE system privilege was originally granted to user `Jim` with the WITH ADMIN OPTION clause, this example revokes the ability to administer the BACKUP DATABASE system privilege from user `Jim`. The ability to perform tasks authorized by the system privilege remains. However, if the BACKUP DATABASE system privilege was originally granted to user `Jim` with the WITH ADMIN ONLY OPTION clause, this example removes all permissions to the system privilege from user `Jim`.

```
REVOKE ADMIN OPTION FOR BACKUP DATABASE FROM Jim
```

Usage

Depending on how the system privilege was initially granted, using the ADMIN OPTION FOR clause when revoking a system privilege has different results. If you the system privilege was originally granted with the WITH ADMIN OPTION clause, including the ADMIN OPTION FOR clause in the revoke statement revokes only the ability to administer the system privilege (that is, grant the system privilege to another user). The ability to actually use the system privilege remains. However, if the system privilege was originally granted with the WITH ADMIN ONLY OPTION clause, including the ADMIN OPTION FOR clause in the revoke statement is semantically equivalent to revoking the entire system privilege. Finally, if the system privilege was originally grant with the WITH NO ADMIN OPTION clause, and the ADMIN OPTION FOR clause is included in the revoke statement, nothing is revoked because there were no administrative system privileges granted in the first place.

Standards

- SQL–other syntaxes are vendor extensions to ISO/ANSI SQL grammar.
- SAP Database Products–syntax is not supported by SAP ASE.

Permissions

Requires administrative privilege over the system privilege being revoked.

In this section:

4.1.29.1 List of All System Privileges

A list of all system privileges.

System privileges control the rights of users to perform authorized database tasks.

The following is a list of available system privileges:

- 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

4.1.30 REVOKE USAGE ON SEQUENCE Statement

Removes USAGE privilege on a specified sequence.

Syntax

```
REVOKE USAGE ON SEQUENCE <sequence-name>
   FROM <userID> [,...]
```

Parameters

- **userID** Must be the name of an existing user or role that has a login password. Separate multiple userIDs with commas.

Standards

- SQL–syntax is a Persistent Stored Module feature.
- SAP Database Products–the security model is different in SAP ASE and SAP IQ, so other syntaxes differ.

Permissions

Requires one of:

- MANAGE ANY OBJECT PRIVILEGE system privilege.
- You own the sequence.

4.1.31 SET OPTION Statement

Changes options that affect the behavior of the database and its compatibility with Transact-SQL. Setting the value of an option can change the behavior for all users or an individual user, in either a temporary or permanent scope.

Syntax

```
SET [ EXISTING ] [ TEMPORARY ] OPTION
   … [ <userid>. | PUBLIC.]<option-name> = [ <option-value> ]
```

Parameters

- **option-value** a host-variable (indicator allowed), string, identifier, or number. The maximum length of `<option-value>` when set to a string is 127 bytes. If `<option-value>` is omitted, the specified option setting is deleted from the database. If it was a personal option setting, the value used reverts to the PUBLIC setting.

> **i Note**
>
> For all database options that accept integer values, SAP IQ truncates any decimal <option-value> setting to an integer value. For example, the value 3.8 is truncated to 3.

- **EXISTING** option values cannot be set for an individual user ID unless there is already a PUBLIC user ID setting for that option.
- **TEMPORARY** changes the duration that the change takes effect. Without the TEMPORARY clause, an option change is permanent: it does not change until it is explicitly changed using `SET OPTION`. When the TEMPORARY clause is applied using an individual user ID, the new option value is in effect as long as that user is logged in to the database.

  When the TEMPORARY clause is used with the PUBLIC user ID, the change is in place for as long as the database is running. When the database is shut down, TEMPORARY options for the PUBLIC user ID revert to their permanent value.

  If a TEMPORARY option is deleted, the option setting reverts to the permanent setting.

Examples

Example 1
set the DATE_FORMAT option:

```
SET OPTION public.date_format = 'Mmm dd yyyy'
```

Example 2
set the WAIT_FOR_COMMIT option to on:

```
SET OPTION wait_for_commit = 'on'
```

Example 3
embedded SQL examples:

```
EXEC SQL SET OPTION :user.:option_name = :value;
EXEC SQL SET TEMPORARY OPTION Date_format = 'mm/dd/yyyy';
```

Usage

The classes of options are:

- General database options
- Transact-SQL compatibility database options

Specifying either a user ID or the `PUBLIC` user ID determines whether the option is set for an individual user, a role represented by <userid>, or the `PUBLIC` user ID (the role to which all users are a member). If the option applies to a role ID, option settings are not inherited by members of the role—the change is applied only to the role ID. If no role is specified, the option change is applied to the currently logged-in user ID that issued the SET OPTION statement. For example, this statement applies an option change to the PUBLIC user ID:

```
SET OPTION Public.login_mode = standard
```

In Embedded SQL, only database options can be set temporarily.

Changing the value of an option for the PUBLIC user ID sets the value of the option for any user that has not set its own value. Option values cannot be set for an individual user ID unless there is already a PUBLIC user ID setting for that option.

Temporarily setting an option for the PUBLIC user ID, as opposed to setting the value of the option permanently, offers a security advantage. For example, when the LOGIN_MODE option is enabled, the database relies on the login security of the system on which it is running. Enabling the option temporarily means a database relying on the security of a Windows domain is not compromised if the database is shut down and copied to a local machine. In that case, the temporary enabling of LOGIN_MODE reverts to its permanent value, which might be Standard, a mode in which integrated logins are not permitted.

> ⚠ Caution
>
> Changing option settings while fetching rows from a cursor is not supported, as it can lead to unpredictable behavior. For example, changing the DATE_FORMAT setting while fetching from a cursor returns different date formats among the rows in the result set. Do not change option settings while fetching rows.

Standards

- SQL—Vendor extension to ISO/ANSI SQL grammar.
- SAP Database Products—Not supported by SAP ASE. SAP IQ does support some SAP ASE options using the SET statement.

Permissions

No specific system privileges are required to set your own options.

The SET ANY PUBLIC OPTION system privilege is required to set database options for another user.

The SET ANY SYSTEM OPTION system privilege is required to set a SYSTEM option for the PUBLIC user ID.

The SET ANY SECURITY OPTION system privilege is required to set a SECURITY option for the PUBLIC user ID.

4.1.32 SETUSER Statement

Allows a user to temporarily assume the roles and system privileges of another user (also known as impersonation) to perform operations, provided they already have the minimum required privileges to perform the task to begin with.

> ℹ Note
>
> The SET USER system privilege is two words; the SETUSER statement is one word.

Syntax

```
SETUSER <userID>
```

Parameters

- **UserID** must be the name of an existing user or role that has a login password.

Usage

At-least criteria validation occurs when the SETUSER statement is executed, not when the SET USER system privilege is granted.

To terminate a successful impersonation, issue the SETUSER statement without specifying a userID.

Standards

ANSI SQL–Compliance level: Transact-SQL extension.

Permissions

Requires the following:

- The impersonator has been granted the right to impersonate the target user.
- The impersonator has, at minimum, all the roles and system privileges granted to the target user.
- The impersonator has been granted the said roles and system privileges with similar or higher administrative rights.

> **i Note**
>
> For the purposes of meeting administrative rights criteria, the WITH ADMIN OPTION and WITH ADMIN ONLY OPTION clauses are considered to grant similar administrative rights. They are also considered to grant higher administrative rights than the WITH NO ADMIN OPTION clause. For example, `User1` is granted `Role1` with the WITH ADMIN OPTION clause, `User2` is granted `Role1` with the WITH ADMIN ONLY clause, and `User3` is granted `Role1` with the WITH NO ADMIN OPTION clause. `User1` and `User2` are said to be granted `Role1` with similar administrative rights. `User1` and `User2` are also said to be granted `Role1` with higher administrative rights than `User3`.

- If the target user has been granted a system privilege that supports extensions, the clauses used to grant the system privilege to the impersonator are a super-set of those used for the target user. Only the SET USER and CHANGE PASSWORD system privileges support extensions.
  - The ANY clause is considered a super-set of the `<target_roles_list>` and `<target_users_list>` clauses. If the target user has been granted the SET USER system privilege with an ANY grant, the impersonator must also have the ANY grant.
  - If the target user has been granted the SET USER system privilege with both the `<target_roles_list>` and `<target_users_list>` clauses, the impersonator must also have been granted the system privilege with the two clauses, and the target list of each clause must be equal to, or a super set of, the corresponding clause grant of the target user. For example, if the target lists of both the impersonator and target user contain `User1`, `User2` and `Role1`, `Role2`, respectively, the target list grants for each clause are said to be equal. Alternately, if the target list grants of the impersonator contain `User1`, `User2`, and `Role1`, `Role2`, respectively, while the target list grants of the target user contain `User1` and `Role2` only, the target list grants of the impersonator are said to be a super-set of the target user.
  - If the target user has been granted the SET USER system privilege with a single target list clause, the target list of the impersonator must be equal to or a super-set of the list of the target user. For example, the `<target_user_list>` of both the impersonator and the target user contain `User1` and `User2` (equal) or the impersonator list contains `User1`, `User2`, while the target user contains `User2`; `User1`, `User2` (impersonator list) is a super-set of `User2` (target user list).
  - By definition, a user can always impersonate himself or herself. Therefore, if the target user is granted the right to impersonate the impersonator, this does not violate the equal to or a super-set of criteria requirement of the impersonator. For example, `User3` is the impersonator and `User4` is the target user. The `<target_user_list>` for `User3` contains `User4` and `User5`. The `<target_user_list>` for `User4` contains `User3` and `User5`. If you remove the impersonator from the target list, the target list of `User3` meets the criteria requirement.

Related Information

Stopping Impersonation of Another User [page 143]

4.1.33 VALIDATE LDAP SERVER Statement

Validates changes to the settings of existing LDAP server configuration objects before applying them.

Syntax

```
VALIDATE LDAP SERVER [ <ldapua-server-name> | ldapua-server-attribs  ]
   [ CHECK <userid> [ <user-dn-string> ] ]
ldapua-server-attribs
   SEARCH DN
            URL { '<URL_string>' | NULL }
            | ACCESS ACCOUNT { '<DN_string>' | NULL }
            | IDENTIFIED BY ( '<password>' | NULL }
            | IDENTIFIED BY ENCRYPTED { <encrypted-password> | NULL }
   | AUTHENTICATION URL { '<URL_string>' | NULL }
   | CONNECTION TIMEOUT <timeout_value>
   | CONNECTION RETRIES <retry_value>
   | TLS { ON | OFF }
```

Parameters

- **ldapua-server-name** identifies the LDAP server configuration object.
- **URL** identifies the host (by name or by IP address), port number, and the search to be performed for the DN lookup for a given user ID. This value is validated for correct LDAP URL syntax before it is stored in the `ISYSLDAPSERVER` system table. The maximum size for this string is 1024 bytes.
- **ACCESS ACCOUNT** a user created on the LDAP server for use by SAP IQ, not a user within SAP IQ. The distinguished name (DN) for this user is used to connect to the LDAP server. This user has permissions within the LDAP server to search for DNs by user ID in the locations specified by the SEARCH DN URL. The maximum size for this string is 1024 bytes.
- **IDENTIFIED BY** provides the password associated with the ACCESS ACCOUNT user. The password is stored using symmetric encryption on disk. Use the value NULL to clear the password and set it to none. The maximum size of a clear text password is 255 bytes.
- **IDENTIFIED BY ENCRYPTED** configures the password associated with the ACCESS ACCOUNT distinguished name in an encrypted format. The binary value is the encrypted password and is stored on disk as is. Use the value NULL to clear the password and set it to none. The maximum size of the binary is 289 bytes.
- **AUTHENTICATION URL** identifies the host (by name or IP address) and the port number of the LDAP server to use for authentication of the user. This is the value defined for <URL_string> and is validated for correct LDAP URL syntax before it is stored in `ISYSLDAPSERVER` system table. The DN of the user obtained from a prior DN search and the user password bind a new connection to the authentication URL. A successful connection to the LDAP server is considered proof of the identity of the connecting user. The maximum size for this string is 1024 bytes.
- **CONNECTION TIMEOUT** specifies the connection timeout from SAP IQ to the LDAP server for both DN searches and authentication. This value is in milliseconds, with a default value of 10 seconds.
- **CONNECTION RETRIES** specifies the number of retries on connections from SAP IQ to the LDAP server for both DN searches and authentication. The valid range of values is 1 – 60, with a default value of 3.

- **TLS** defines whether the TLS or Secure LDAP protocol is used for connections to the LDAP server for both DN searches and authentication. When set to ON, the TLS protocol is used and the URL begins with "ldap://" When set to OFF (or not specified), Secure LDAP protocol is used and the URL begins with “ldaps://”. When using the TLS protocol, specify the database security option TRUSTED_CERTIFICATES_FILE with a file name containing the certificate of the Certificate Authority (CA) that signed the certificate used by the LDAP server.
- **CHECK userID** the userID whose existence is validated on the LDAP server.
- **user-dn-string** compares a user's DN value with the user ID for verification purposes.

Examples

Example 1

assume the apps_primary LDAP server configuration object was created as follows:

```
SET OPTION PUBLIC.login_mode = 'Standard,LDAPUA'
CREATE LDAP SERVER apps_primary
SEARCH DN
    URL 'ldap://my_LDAPserver:389/dc=MyCompany,dc=com??sub?cn=*'
    ACCESS ACCOUNT 'cn=aseadmin, cn=Users, dc=mycompany, dc=com'
    IDENTIFIED BY 'Secret99Password'
AUTHENTICATION URL 'ldap://my_LDAPserver:389/'
CONNECTION TIMEOUT 3000
WITH ACTIVATE
```

This statement validates the existence of a userID `myusername` by using the optional CHECK clause to compare the userID to the expected user distinguished name (enclosed in quotation marks) on the `apps_primary` LDAP server configuration object.

```
VALIDATE LDAP SERVER apps_primary
CHECK myusername 'cn=myusername,cn=Users,dc=mycompany,dc=com'
```

Example 2

the name of the LDAP server configuration object does not have to defined in the `VALIDATE LDAP SERVER` statement if you include the search attributes:

```
VALIDATE LDAP SERVER
SEARCH DN
    URL  'ldap://my_LDAPserver:389/dc=MyCompany,dc=com??sub?cn=*'
    ACCESS ACCOUNT 'cn=aseadmin, cn=Users, dc=mycompany, dc=com'
    IDENTIFIED BY 'Secret99Password'
AUTHENTICATION URL 'ldap://my_LDAPserver:389/'
CONNECTION TIMEOUT 3000
CHECK myusername 'cn=myusername,cn=Users,dc=mycompany,dc=com'
```

Usage

This statement is useful for an administrator when setting up a new server to use LDAP user authentication, and for diagnosing problems between the LDAP server configuration object and the external LDAP server. Any connection made by the `VALIDATE LDAP SERVER` statement is temporary and is closed by the end of the statement.

When validating the LDAP server configuration object by name, definitions from prior `CREATE LDAP SERVER` and `ALTER LDAP SERVER` statements are used. Alternately, when `<ldapua-server-attributes>` are specified instead of the LDAP server configuration object name, the specified attributes are validated. When `<ldapua-server-attributes>` are specified, the URLs are parsed to identify syntax errors, and statement processing stops is a syntax error is detected.

Whether using an LDAP server configuration object name or a successfully parsed set of `<ldapua-server-attributes>`, a connection to the external LDAP server is attempted. If the parameter ACCESS ACCOUNT and a password are specified, the values are used to establish the connection to the SEARCH DN URL. This is the SEARCH DN URL, ACCESS ACCOUNT, and ACCESS ACCOUNT password.

When using the optional CHECK clause, the userID is used in the search to validate the existence of the user on the external LDAP server. When the expected DN value for a given user is known, the value can be specified, and is compared with the result of the search to determine success or failure.

Standards

ANSI SQL–Compliance level: Transact-SQL extension.

Permissions

Requires the MANAGE ANY LDAP SERVER system privilege.

4.2 Database Options

Database options customize and modify database behavior.

In this section:

Extends LOGIN_MODE for LDAPUA only to a select number of users using Standard authentication.

VERIFY_PASSWORD_FUNCTION Option [page 359]
Specifies a user-supplied authentication function that can be used to implement password rules.

MIN_PASSWORD_LENGTH Option [page 362]
Sets the minimum length for new passwords in the database.

-gk database server option [page 363]
Sets the privileges required to stop the database server.

-gl database server option [page 364]
Set the permission required to load data using `LOAD TABLE`.

-gu database server option [page 364]
Sets the privilege required for executing database file administration statements such as for creating or dropping databases.

-sk database server option [page 366]
Specifies a system secure feature key that can be used to allow access to features that are secured for the database server.

-sf database server option [page 367]
Controls whether users have access to features for databases running on the current database server.

4.2.1 LOGIN_MODE Option

Controls the use of standard, integrated, Kerberos, LDAP, and PAM logins for the database.

Allowed Values

- Standard – the default setting, which does not permit integrated logins. An error occurs if an integrated login connection is attempted.
- Mixed – allows both integrated logins and standard logins.
- Integrated – all logins to the database must be made using integrated logins.
- Kerberos – all logins to the database must be made using Kerberos logins.
- LDAPUA – all logins to the database must be made using LDAP logins.
- PAMUA – all logins to the database must be made using PAM logins.

> i Note
>
> Mixed is equivalent to "Standard,Integrated."

Default

Standard

Scope

Option can be set at the database (PUBLIC) level only.

Requires the SET ANY SECURITY OPTION system privilege to set this option. Takes effect immediately.

Remarks

Values are case-insensitive. Specify values in a comma-separated list without white space.

> ⚠ Caution
>
> - Restricting the `LOGIN_MODE` to a single mode in a mixed environment (for example, integrated only or LDAPUA only) restricts connections to only those users who have been granted the corresponding login mapping. Attempting to connect using other methods generates an error. The only exceptions to this are users with full administrative rights (SYS_AUTH_DBA_ROLE or SYS_AUTH_SSO_ROLE).
> - Restricting the `LOGIN_MODE` to LDAPUA only may result in a configuration where no users can connect to the server if no user or login policy exists that permits LDAPUA. Use the command line switch `-al <user-id-list>` with the `start_iq` utility to recover from this situation.
> - If a database file is not secured and can be copied by unauthorized users, set the LOGIN_MODE as a TEMPORARY public option for integrated, Kerberos, or PAM user authentication. This ensures that, by default, integrated, Kerberos, and PAM logins are not supported if the file is copied.

4.2.2 MIN_ROLE_ADMINS Option

Configures of the minimum number of required administrators for all roles.

Allowed Values

1 – 10

Default

1

Scope

Option can be set at the database (PUBLIC) level only.

Requires the SET ANY SECURITY OPTION system privilege to set this option. Takes effect immediately.

Remarks

This option sets the minimum number of required administrators for all roles. This value applies to the minimum number of role administrators for each role, not the minimum number or role administrators for the total number of roles. When dropping roles or users, this value ensures that you never create a scenario where there are no users and roles left with sufficient system privilege to manage the remaining users and roles.

4.2.3 TRUSTED_CERTIFICATES_FILE Option

Specifies the trust relationship for outbound Transport Layer Security (TLS) connections made by LDAP User Authentication, INC, DAS INC, and MIPC connections.

Allowed Values

A valid network path to the location of a TXT file containing the list of trusted certificate authorities that sign server certificates.

Default

NULL, meaning that no outbound TLS connection can be started because there are no trusted certificate authorities.

Scope

Option can be set at the database (PUBLIC) level only.

Requires the SET ANY SECURITY OPTION system privilege to set this option. Takes effect immediately.

Remarks

This option identifies the path to the location of the list of trusted certificate authorities. The list must be stored in a TXT file. The file may be shared in a location in a Windows environment on the local drive to be used by all SAP applications on that machine.

4.2.4 -al database server option

Extends LOGIN_MODE for LDAPUA only to a select number of users using Standard authentication.

Syntax

```
start_iq -al <"user1;user2;user3" server_name.cfg database-name.db
>
```

Remarks

- Up to five user IDs can be specified, separated by semi-colons, and enclosed in double quotation marks.
- When run at the server level, the `-al` switch remains in effect until the next time the server is restarted.

4.2.5 -al database option

Extends LOGIN_MODE for LDAPUA only to a select number of users using Standard authentication.

Syntax

```
start_iq -al <"user1;user2;user3" server_name.cfg database_name.db>
```

Remarks

- Up to five user IDs can be specified, separated by semi-colons, and enclosed in double quotation marks.
- When run a t the database level, it remains in effect until the next time the database is stopped/started.

4.2.6 VERIFY_PASSWORD_FUNCTION Option

Specifies a user-supplied authentication function that can be used to implement password rules.

Allowed Values

String

Default

'' (the empty string). (No function is called when a password is set.)

Scope

Option can be set at the database (PUBLIC) or user level. At the database level, the value becomes the default for any new user, but has no impact on existing users. At the user level, overrides the PUBLIC value for that user only. No system privilege is required to set option for self. System privilege is required to set at database level or at user level for any user other than self.

Requires the SET ANY SECURITY OPTION system privilege to set this option. Can be set temporary for an individual connection or for the PUBLIC role. Takes effect immediately.

Remarks

When the `VERIFY_PASSWORD_FUNCTION` option value is set to a valid string, the statement `GRANT CONNECT TO <userid> IDENTIFIED BY <password>` calls the function specified by the option value.

The option value requires the form `<owner.function_name>` to prevent users from overriding the function.

The function takes two parameters:

- `<user_name>` `VARCHAR`(128)
- `<new_pwd>` `VARCHAR`(255)

The return value type is `VARCHAR`(255).

If `VERIFY_PASSWORD_FUNCTION` is set, you cannot specify more than one userid and password with the `GRANT CONNECT` statement.

Example

The following sample code defines a table and a function and sets some login policy options. Together they implement advanced password rules that include requiring certain types of characters in the password, disallowing password reuse, and expiring passwords. The function is called by the database server with the `VERIFY_PASSWORD_FUNCTION` option when a user ID is created or a password is changed. The application can call the procedure specified by the `POST_LOGIN_PROCEDURE` option to report that the password should be changed before it expires.

```
-- THIS EXAMPLE ONLY WORKS WITH A SINGLE BYTE COLLATION DATABASE.
-- The example checks for alpha-numeric characters in the password.
-- To allow or check for other characters, such as "*" or "%", customize the
example.
--
-- Only the DBA should have privileges on this table.
```

```
CREATE TABLE DBA.t_pwd_history(
        pk          INT         DEFAULT AUTOINCREMENT PRIMARY KEY,
        user_name   CHAR(128),  -- the user whose password is set
        pwd_hash    CHAR(32) ); -- hash of password value to detect
                                -- duplicate passwords
-- called whenever a non-NULL password is set
-- to verify the password conforms to password rules
CREATE FUNCTION DBA.f_verify_pwd( uid     VARCHAR(128),
                                  new_pwd VARCHAR(255) )
RETURNS VARCHAR(255)
BEGIN
    -- a table with one row per character in new_pwd
    DECLARE local temporary table pwd_chars(
            pos INT PRIMARY KEY,    -- index of c in new_pwd
            c   CHAR( 1 ) );        -- USE BYTE-LENGTH SEMANTICS

    -- new_pwd with non-alpha characters removed
    DECLARE pwd_alpha_only      CHAR(255);
    DECLARE num_lower_chars     INT;
    -- enforce minimum length (can also be done with
    -- min_password_length option)
    IF length( new_pwd ) < 6 THEN
        RETURN 'password must be at least 6 characters long';
    END IF;
    -- break new_pwd into one row per character
    INSERT INTO pwd_chars SELECT row_num, substr( new_pwd, row_num, 1 )
                            FROM dbo.RowGenerator
                            WHERE row_num <= length( new_pwd );
    -- copy of new_pwd containing alpha-only characters
    SELECT list( c, '' ORDER BY pos ) INTO pwd_alpha_only
        FROM pwd_chars WHERE c BETWEEN 'a' AND 'z' OR c BETWEEN 'A' AND 'Z';
    -- number of lowercase characters IN new_pwd
    SELECT count(*) INTO num_lower_chars
        FROM pwd_chars WHERE CAST( c AS BINARY ) BETWEEN 'a' AND 'z';
    -- enforce rules based on characters contained in new_pwd
    IF ( SELECT count(*) FROM pwd_chars WHERE c BETWEEN '0' AND '9' )
           < 1 THEN
        RETURN 'password must contain at least one numeric digit';
    ELSEIF length( pwd_alpha_only ) < 2 THEN
        RETURN 'password must contain at least two letters';
    ELSEIF num_lower_chars = 0
           OR length( pwd_alpha_only ) - num_lower_chars = 0 THEN
        RETURN 'password must contain both upper- and lowercase characters';
    END IF;
    -- not the same as any user name
    -- (this could be modified to check against a disallowed words table)
    IF EXISTS( SELECT * FROM SYS.SYSUSER
                    WHERE lower( user_name ) IN ( lower( pwd_alpha_only ),
                                                  lower( new_pwd ) ) ) THEN
        RETURN 'password or only alphabetic characters in password ' ||
               'must not match any user name';
    END IF;
    -- not the same as any previous password for this user
    IF EXISTS( SELECT * FROM t_pwd_history
                    WHERE user_name = uid
                      AND pwd_hash = hash( uid || new_pwd, 'md5' ) ) THEN
        RETURN 'previous passwords cannot be reused';
    END IF;
    -- save the new password
    INSERT INTO t_pwd_history( user_name, pwd_hash )
        VALUES( uid, hash( uid || new_pwd, 'md5' ) );
    RETURN( NULL );
END;
ALTER FUNCTION DBA.f_verify_pwd SET HIDDEN;
GRANT EXECUTE ON DBA.f_verify_pwd TO PUBLIC;
SET OPTION PUBLIC.verify_password_function = 'DBA.f_verify_pwd';
-- All passwords expire in 180 days. Expired passwords can be changed
-- by the user using the NewPassword connection parameter.
```

```
ALTER LOGIN POLICY DEFAULT password_life_time = 180;
-- If an application calls the procedure specified by the
-- post_login_procedure option, then the procedure can be used to
-- warn the user that their password is about to expire. In particular,
-- Interactive SQL calls the post_login_procedure.
ALTER LOGIN POLICY DEFAULT password_grace_time = 30;
-- Five consecutive failed login attempts results in a non-DBA
-- user ID being locked.
ALTER LOGIN POLICY DEFAULT max_failed_login_attempts = 5;
```

To turn the option off, set it to the empty string:

```
SET OPTION PUBLIC.VERIFY_PASSWORD_FUNCTION = ''
```

4.2.7 MIN_PASSWORD_LENGTH Option

Sets the minimum length for new passwords in the database.

Allowed Values

Integer greater than or equal to zero

The value is in bytes. For single-byte character sets, this is the same as the number of characters.

Default

3 characters

Scope

Option can be set at the database (PUBLIC) level only.

Requires the SET ANY SECURITY OPTION system privilege to set this option. Takes effect immediately.

Remarks

This option imposes a minimum length on all new passwords for greater security. Existing passwords are not affected.

Example

Set the minimum length for new passwords to 6 bytes:

```
SET OPTION PUBLIC.MIN_PASSWORD_LENGTH = 6
```

4.2.8 -gk database server option

Sets the privileges required to stop the database server.

Syntax

```
start_iq -gk { DBA | all | none } ...
```

Allowed values

DBA

Only users with the SERVER OPERATOR system privilege can stop the database server. This is the default for the network server.

all

No privileges are required to shut down the database server.

none

The database server cannot be stopped.

Applies to

All operating systems and database servers.

Remarks

The -gd database server option applies to the dbstop utility as well as to the following statements:

- ALTER DATABASE `<dbname>` FORCE START statement.
- STOP DATABASE statement

4.2.9 -gl database server option

Set the permission required to load data using LOAD TABLE.

Syntax

```
start_iq -gl <level>
```

Remarks

The LOAD TABLE statement reads files from the database server machine

. To control access to the file system using these statements, the -gl <level> command-line swicodetch allows you to control the level of database privileges that are required to use these statements, as follows:

Level	Description
DBA	Only users with the LOAD ANY TABLE, ALTER ANY TABLE or ALTER ANY OBJECT system privilege can load data.
ALL	Only users with one of the following: • You are the owner of the table • ALTER object-level privilege on the table • LOAD object-level privilege on the table • ALTER ANY TABLE system privilege • LOAD ANY TABLE system privilege • ALTER ANY OBJECT system privilege
NONE	Data cannot be loaded.

The option is case insensitive.

The default setting is <ALL> for servers started with start_iq and <DBA> for all other servers. For consistency with earlier versions, use <ALL> on all systems. <ALL> is used in the iqdemo.cfg and default.cfg configuration files.

4.2.10 -gu database server option

Sets the privilege required for executing database file administration statements such as for creating or dropping databases.

Syntax

```
start_iq -gu { all | none | DBA | utility_db } ...
```

Allowed values

-gu option	Effect	Applies to
all	*This option is deprecated.* Anyone can execute file administration statements.	Any database including utility database
none	Executing file administration statements is not allowed.	Any database including utility database
DBA	Only users with the SERVER OPERATOR system privilege can execute file administration statements	Any database including utility database
utility_db	Only the users who can connect to the utility database can execute file administration statements	Only the utility database

Default

DBA

Remarks

Restricts the users who can execute the following database file administration statements:

- ALTER DATABASE dbfile ALTER TRANSACTION LOG
- CREATE DATABASE statement
- DROP DATABASE statement
- RESTORE DATABASE statement.

When utility_db is specified, these statements can only be run from the utility database. When DBA is specified, these statements can only be run by a user with the SERVER OPERATOR system privilege. When none is specified, no user can execute these statements.

Example

To prevent the use of the file administration statements, start the database server using the none privilege level of the -gu option. The following command starts a database server and names it TestSrv. It loads the `mytestdb.db` database, but prevents anyone from using that server to create or delete a database, or execute any other file administration statement regardless of their resource creation rights, or whether they can load and connect to the utility database.

```
start_iq -n TestSrv -gu none c:\mytestdb.db
```

To permit only the users knowing the utility database password to execute file administration statements, start the server by running the following command.

```
start_iq -n TestSrv -su secret -gu utility_db
```

The following command starts Interactive SQL as a client application, connects to the server named TestSrv, loads the utility database, and connects the user.

```
dbisql -c "UID=DBA;PWD=secret;DBN=utility_db;Host=host1;Server=TestSrv"
```

Having executed the above command successfully, the user connects to the utility database, and can execute file administration statements.

4.2.11 -sk database server option

Specifies a system secure feature key that can be used to allow access to features that are secured for the database server.

Syntax

```
start_iq -sk <key> ...
```

Applies to

All operating systems and database servers.

Remarks

When you secure features for a database server by using the -sf option, you can also include the -sk option, which specifies a key that can be used with the sp_use_secure_feature_key system procedure to allow access to secured features for a connection. That connection can also use the sa_server_option system procedure to modify the features or feature sets that are secured for all databases running on the database server.

The key must be a non-empty string of at least six characters, and it cannot contain double quotes, control characters (any character less than 0x20), or backslashes. There is a limit of 1000 secure feature keys per database.

If the value for the authorization_key parameter of the sp_use_secure_feature_key system procedure is set to any value other than the one specified by -sk, no error is given and the features specified by -sf remain secured for the connection.

If you specify -sk without -sf, only the default secure features are enabled, but you can use the system secure feature key while the database server is running to change the secure feature settings.

Example

The following command starts a database server named secure_server with the backup feature secured. The key specified by the -sk option can be used later to allow access to these features for a specific connection.

```
start_iq -n secure_server -sf backup -sk j978kls12
```

Setting the authorization_key parameter to the value specified by -sk for a connection to a database running on the secure_server database server allows that connection to perform backups or change the features that are secured on the secure_server database server:

```
CALL sp_use_secure_feature_key ( 'MyKey' , 'j978kls12' );
```

The user can then secure all features for databases running on secure_server by executing the following statement:

```
CALL sa_server_option( 'SecureFeatures', 'all' );
```

4.2.12 -sf database server option

Controls whether users have access to features for databases running on the current database server.

A secured feature can only be accessed by a user with appropriate privileges, while an unsecured feature can be accessed by all users.

Syntax

```
start_iq -sf <feature-list> ...
```

```
<feature-list> :
<feature-name> | <feature-set> [ ,<feature-name> | <feature-set> ] ...
```

Feature set	Included features (feature sets in bold)
none	All features are unsecured except manage_features, manage_keys, and disk_sandbox.
manage_server	• processor_affinity
manage_security	• manage_features • manage_keys • manage_disk_sandbox
server_security	• disk_sandbox • trace_system_event

Feature set	Included features (feature sets in bold)
all	**client** • read_client_file • write_client_file **remote** • remote_data_access • send_udp • send_email • web_service_client **local** **local_call** • cmdshell • external_procedure_v3-External C or C++ procedure. See the User-Defined Functions guide for information on C and C++ UDFs. • java **local_db** • backup • restore • database • dbspace **local_env** • getenv **local_io** • create_trace_file • read_file • write_file • directory • sp_list_directory • sp_create_directory • sp_copy_directory • sp_move_directory • sp_delete_directory • sp_copy_file • sp_move_file • sp_delete_file **local_log** • request_log • console_log • webclient_log

Parameters

none

Specifies that no features are secured.

manage_server

Prevents users from accessing all database server-related features. This set consists of the following features:

> **processor_affinity**
>
> Prevents users from changing the processor affinity (the number of logical processors being used) of the database server.

manage_security

Prevents users from accessing features that allow the management of database server security. By default, these features are secured.

> **manage_features**
>
> Prevents users from modifying the list of features that can be secured on the database server.
>
> **manage_keys**
>
> Prevents the creation, modification, deletion, or listing of secure feature keys.
>
> A user that has access to the manage_keys feature but not the manage_features feature cannot define a key with more secure features than those assigned to the user.
>
> **manage_disk_sandbox**
>
> Prevents users from temporarily changing disk sandbox settings by using the sa_server_option system procedure or the sa_db_option system procedure. The manage_disk_sandbox secure feature cannot be turned off for all databases or users: it can only be turned off for individual connections by using the sp_use_secure_feature_key system procedure.

server_security

Prevents users from accessing features that can temporarily bypass security settings. By default, these features are secured.

> **disk_sandbox**
>
> Prevents users from performing read-write file operations on the database outside the directory where the main database file is located.
>
> **trace_system_event**
>
> Prevents users from creating user-defined trace events.

all

Prevents users from accessing the following groups:

> **client**
>
> Prevents users from accessing all features that allow access to client-related input and output. This feature controls access to the client computing environment. This set consists of the following features:
>
> > **read_client_file**
> >
> > Prevents the use of statements that can cause a client file to be read. For example, the READ_CLIENT_FILE function and the LOAD TABLE statement.
> >
> > **write_client_file**

Prevents the use of all statements that can cause a client file to be written to. For example, the UNLOAD statement and the WRITE_CLIENT_FILE function.

remote

Prevents users from accessing all features that allow remote access or communication with remote processes. This set consists of the following features:

remote_data_access

Prevents the use of any remote data access services, such as proxy tables.

send_udp

Prevents the ability to send UDP packets to a specified address by using the sa_send_udp system procedure.

send_email

Prevents the use of email system procedures, such as xp_sendmail.

web_service_client

Prevents the use of web service client stored procedure calls (stored procedures that issue HTTP requests).

local

Prevents users from accessing all local-related features. This feature controls access to the server computing environment. This set consists of the local_call, local_db, local_io, and local_log feature subsets.

local_call

Prevents users from accessing all features that provide the ability to execute code that is not directly part of the database server and is not controlled by the database server. This set consists of the following features:

cmdshell

Prevents the use of the xp_cmdshell procedure.

external_procedure

Prevents the use of external stored procedures. This setting does not disable the use of the xp_* system procedures (such as xp_cmdshell, xp_readfile, and so on) that are built into the database server. Separate feature control options are provided for these system procedures.

external_procedure_v3

See the User-Defined Functions guide.

java

Prevents the use of Java-related features, such as Java procedures.

local_db

Prevents users from accessing all features related to database files. This set consists of the following features:

backup

Prevents the use of the BACKUP DATABASE statement, and with it, the ability to run server-side backups. You can still perform client-side backups by using the dbbackup utility.

restore

Prevents the use of the RESTORE DATABASE statement.

database

Prevents the use of the CREATE DATABASE, ALTER DATABASE, and DROP DATABASE statements.

dbspace

Prevents the use of the CREATE DBSPACE, ALTER DBSPACE, and DROP DBSPACE statements.

local_env

Prevents users from accessing all features related to environment variables. This set consists of the following features:

getenv

Prevents users from reading the value of any environment variable.

local_io

Prevents users from accessing all features that allow direct access to files and their contents. This set consists of the following features:

create_trace_file

Prevents the use of statements that create an event tracing target.

read_file

Prevents the use of statements that can cause a local file to be read. For example, the xp_read_file system procedure, the LOAD TABLE statement, and the use of OPENSTRING(FILE...). The alternate names load_table and xp_read_file are deprecated.

write_file

Prevents the use of all statements that can cause a local file to be written to. For example, the UNLOAD statement and the xp_write_file system procedure. The alternate names unload_table and xp_write_file are deprecated.

delete_file

Prevents the use of all statements that can cause a local file to be deleted. For example, securing this feature causes the dbbackup utility to fail if the -x or -xo options are specified.

directory

Prevents the use of directory class proxy tables. This feature is disabled when remote_data_access is disabled.

sp_list_directory

Prevents the use of the sp_list_directory system procedure.

sp_create_directory

Prevents the use of the sp_create_directory system procedure.

sp_copy_directory

Prevents the use of the sp_copy_directory system procedure.

sp_move_directory

Prevents the use of the sp_move_directory system procedure.

sp_delete_directory

Prevents the use of the sp_delete_directory system procedure.

sp_copy_file

Prevents the use of the sp_copy_file system procedure.

sp_move_file

Prevents the use of the sp_move_file system procedure.

sp_delete_file

Prevents the use of the sp_delete_file system procedure.

local_log

Prevents users from accessing all logging features that result in creating or writing data directly to a file on disk. This set consists of the following features:

request_log

Prevents the ability to change the request log file name and also prevents the ability to increase the limits of the request log file size or number of files. You can specify the request log file and limits on this file in the command to start the database server; however, they cannot be changed once the database server is started. When request log features are disabled, you can still turn request logging on and off and reduce the maximum file size and number of request logging files.

console_log

Prevents the ability to change the database server message log file name using the ConsoleLogFile option of the sa_server_option system procedure. Securing this feature also prevents the ability to increase the maximum size of the database server message log file using the ConsoleLogMaxSize option of the sa_server_option system procedure. You can specify a server log file and its size when starting the database server.

webclient_log

Prevents the ability to change the web service client log file name using the WebClientLogFile option of the sa_server_option system procedure. You can specify a web service client log file when starting the database server.

Applies to

All operating systems and database servers.

Remarks

This option allows the owner of the database server to control whether users have access to features for databases running on the database server. The -sk option allows the owner of the database server to create a system secure feature key that prevents users from accessing features specified by the -sf option.

If you start a database without specifying a system secure feature key, the default secure features are secured, and you cannot change the secure feature settings for the database server or any databases running on it. You cannot create the system secure feature key later: you must shut down the database server and specify a system secure feature key when you restart it.

The `<feature-list>` is a comma-separated list of feature names or feature sets to secure for the database server. Securing a feature makes it inaccessible to all database users other than administrators. Specifying a feature set secures all the features included in the set. To secure one or more, but not all, of the features in the feature set, specify the individual feature name.

> Note
>
> Sub-features of feature sets that are secured by default, cannot be unsecured from the command line. In other words the following command will not work:
>
> ```
> -sf manage_security, -manage_keys
> ```

Use `<feature-name>` to indicate that the feature should be secured (made inaccessible), and - `<feature-name>` or `<feature-name>` - to indicate that the feature should be unsecured (accessible to all database users). For example, the following command indicates that only dbspace features are accessible to all users:

```
start_iq -n secure_server -sf all,-dbspace
```

Example

The following command starts a database server named secure_server with access to the request log and with all remote data access features secured. The key specified by the -sk option can be used later with the sp_use_secure_feature_key system procedure to make these features accessible to all users on the current connection.

```
start_iq -n secure_server -sf remote,-request_log -sk j978kls12
```

If a user connected to a database running on the secure_server database server uses the sp_use_secure_feature_key system procedure with the authorization_key parameter set to the same value as that specified by -sk, that connection has access to the remote data access features:

```
CALL sp_use_secure_feature_key ( 'MyKey' , 'j978kls12' );
```

The following command secures all features, with the exception of local database features:

```
start_iq -n secure_server -sf all,-local_db
```

4.3 Procedures and Functions

Use the system-supplied stored functions and procedures in SAP IQ databases to retrieve system information.

In this section:

sa_get_ldapserver_status System Procedure [page 375]

Determines the current status of the LDAP server configuration object.

sa_get_user_status system procedure [page 376]

Allows you to determine the current status of users.

sp_displayroles System Procedure [page 378]

Displays all roles granted to a user-defined role or a user, or displays the entire hierarchical tree of roles.

sp_expireallpasswords System Procedure [page 381]

Immediately expires all user passwords.

SP_HAS_ROLE Function [System] [page 382]

Returns an integer value indicating whether the invoking user has been granted a specified system privilege or user-defined role. When used for privilege checking within user-defined stored procedures, `SP_HAS_ROLE` returns an error message when a user fails a privilege check.

sp_iqaddlogin Procedure [page 384]

Adds a new SAP IQ user account to the specified login policy.

sp_iqbackupdetails Procedure [page 386]

Shows all the dbfiles included in a particular backup.

sp_iqbackupsummary Procedure [page 388]

Summarizes backup operations performed.

sp_iqconnection Procedure [page 389]

Shows information about connections and versions, including which users are using temporary dbspace, which users are keeping versions alive, what the connections are doing inside SAP IQ, connection status, database version status, and so on.

sp_iqcopyloginpolicy Procedure [page 392]

Creates a new login policy by copying an existing one.

sp_iqdbspace Procedure [page 393]

Displays detailed information about each SAP IQ dbspace.

sp_iqdbspaceinfo Procedure [page 396]

Displays the size of each object and subobject used in the specified table. Not supported for RLV dbspaces.

sp_iqdbspaceobjectinfo Procedure [page 399]

Lists objects and subobjects of type table (including columns, indexes, metadata, primary keys, unique constraints, foreign keys, and partitions) for a given dbspace. Not supported for RLV dbspaces.

sp_iqdroplogin Procedure [page 402]

Drops an SAP IQ user account.

sp_iqemptyfile Procedure [page 403]

Empties a dbfile and moves the objects in the dbfile to another available read-write dbfile in the same dbspace. Not available for files in an RLV dbspace.

sp_iqestdbspaces Procedure [page 406]

Estimates the number and size of dbspaces needed for a given total index size.

sp_iqfile Procedure [page 407]

Displays detailed information about each dbfile in a dbspace.

sp_iqmodifyadmin Procedure [page 411]

Sets an option on a named login policy to a certain value. If no login policy is specified, the option is set on the root policy. In a multiplex, `sp_iqmodifyadmin` takes an optional parameter that is the multiplex server name.

sp_iqmodifylogin Procedure [page 412]
Assigns a user to a login policy.

sp_iqobjectinfo Procedure [page 413]
Returns partitions and dbspace assignments of database objects and subobjects.

sp_iqspaceused Procedure [page 416]
Shows information about space available and space used in the IQ store, IQ temporary store, RLV store, and IQ global and local shared temporary stores.

sp_iqsysmon Procedure [page 418]
Monitors multiple components of SAP IQ, including the management of buffer cache, memory, threads, locks, I/O functions, and CPU utilization.

sp_iqpassword Procedure [page 437]
Changes a user's password.

sp_objectpermission System Procedure [page 439]
Generates a report on object privileges granted to the specified role, or user name, or the object privileges granted on the specified object or dbspace.

sp_sys_priv_role_info System Procedure [page 442]
Generates a report to map a system privilege to the corresponding system role. A single row is returned for each system privilege.

sp_alter_secure_feature_key System Procedure [page 443]
Alters a previously-defined secure feature key by modifying the authorization key and/or the feature list.

sp_create_secure_feature_key System Procedure [page 444]
Creates a new secure feature key.

sp_drop_secure_feature_key System Procedure [page 445]
Deletes a secure feature key.

sp_list_secure_feature_keys System Procedure [page 445]
Returns information about the contents of a directory.

sp_use_secure_feature_key System Procedure [page 446]
Enables an existing secure feature key.

4.3.1 sa_get_ldapserver_status System Procedure

Determines the current status of the LDAP server configuration object.

Syntax

```
sa_get_ldapserver_status()
```

Privileges

You must have EXECUTE privilege on the system procedure.

Remarks

Column Name	Data Type	Description
ldsrv_id	UNSIGNED BIGINT	A unique identifier for the LDAP server configuration object that is the primary key and is used by the login policy to refer to the LDAP server.
ldsrv_name	CHAR(128)	The name assigned to the LDAP server configuration object.
ldsrv_state	CHAR(9)	Read-only state of the LDAP server: 1 – RESET 2 – READY 3 – ACTIVE 4 – FAILED 5 – SUSPENDED A numeric value is stored in system table; a corresponding text value appears in the system view.
ldsrv_last_state_change	TIMESTAMP	Indicates the time the last state change occurred. The value is stored in Coordinated Universal Time (UTC), regardless of the local time zone of the LDAP server.

To see SYSLDAPSERVER column values before a checkpoint occurs and the contents of memory are written to the catalog on disk. The updates to the catalog columns ldsrv_state and ldsrv_last_state_change occur asynchronously during checkpoint to the LDAP server object as the result of an event that changes the LDAP server object state, such as a failed connection due to a failed LDAP directory server. The LDAP server object state reflects the state of the LDAP directory server.

4.3.2 sa_get_user_status system procedure

Allows you to determine the current status of users.

Syntax

```
sa_get_user_status( )
```

Result set

Column name	Data type	Description
user_id	UNSIGNED INTEGER	A unique number identifying the user.
user_name	CHAR(128)	The name of the user.
connections	INTEGER	The current number of connections by this user.
failed_logins	UNSIGNED INTEGER	The number of failed login attempts made by the user.
last_login_time	TIMESTAMP	The local time that the user last logged in.
locked	TINYINT	Indicates if the user account is locked.
reason_locked	LONG VARCHAR	The reason the account is locked.
user_dn	CHAR(1024)	The Distinguished Name (DN) for a user ID connecting to an LDAP server.
user_dn_cached_at	TIMESTAMP	The local time that the DN was stored.
password_change_state	BIT	A value that indicates whether a dual password change is in progress (0=No, 1=Yes). The default is 0.
password_change_first_user	UNSIGNED INTEGER	The user_id of the user who set the first part of a dual password; otherwise NULL.
password_change_second_user	UNSIGNED INTEGER	The user_id of the user who set the second part of a dual password; otherwise NULL.
user_dn	CHAR(1024)	The distinguished name (DN) of the user.
user_dn_cached_at	TIMESTAMP	The date and time the distinguished name was found.

Remarks

This procedure returns a result set that shows the current status of users. In addition to basic user information, the procedure includes a column indicating if the user has been locked out and a column with a reason for the lockout. Users can be locked out for the following reasons: locked due to policy, password expiry, or too many failed attempts.

If the user is authenticated using LDAP User Authentication, the output includes the user's distinguished name and the date and time that the distinguished name was found.

Privileges

You must have EXECUTE privilege on the system procedure.

To view information about other users, you must also have the MANAGE ANY USER system privilege.

Side effects

None

Example

The following example uses the sa_get_user_status system procedure to return the status of database users.

```
CALL sa_get_user_status;
```

4.3.3 sp_displayroles System Procedure

Displays all roles granted to a user-defined role or a user, or displays the entire hierarchical tree of roles.

Syntax

```
sp_displayroles(
   [ <user_role_name> ],
   [ <display_mode> ],
   [ <grant_type> ] )
```

Parameters

user_role_name valid values are:

- A valid system privilege name or system privilege role name
- A valid user-defined role name
- A valid user name

By default, if no argument is specified, the current login user is used.

display_mode valid values are:

EXPAND_UP shows all roles granted the input role or system privilege; that is the role hierarchy tree for the parent levels.

EXPAND_DOWN shows all roles or system privileges granted to the input role or user; that is, the role hierarchy tree for the child levels.

If no argument is specified (default), only the directly granted roles or system privileges appear.

grant_type valid values are:

ALL shows all roles or system privileges granted.

NO_ADMIN shows all roles or system privileges granted with the WITH NO ADMIN OPTION or WITH ADMIN OPTION clause.

ADMIN shows all roles or system privileges granted with the WITH ADMIN OPTION or WITH ADMIN ONLY OPTION clause.

If no argument is specified, **ALL** is used.

Privileges

You must have EXECUTE privilege on the system procedure. To execute this procedure against other users, you must have the MANAGE ROLES system privilege. To execute against a role or system privilege, you must be an administrator of the role or have administrative rights to the system privilege.

Remarks

Column Name	Data Type	Description
role_name	char(128)	Lists role/system privilege name.
parent_role_name	char(128)	Lists role name of the parent.
grant_type	char(10)	Lists grant type.
role_level	smallint	For `Expand_down` mode, 1 indicates directly granted roles; 2 indicates the next hierarchy below, and so on. For `Expand_up` mode, 0 indicates the roles to which the specified role is granted; -1 indicates the next hierarchy above, and so on.

For Name = System privilege name, the results show the system privilege name instead of the system privilege role name.

For Mode = Expand_down, parent_role_name is NULL for level 1 (directly granted roles). If no mode is specified (default), role_level is 1 and parent_role_name is NULL, since only directly granted roles appear.

For Name = User name, with Mode = expand_up, no results are returned since a user resides at the top level in any role hierarchy. Similarly, if Name = an immutable system privilege name, with Mode = Expand_down, no results are returned because an immutable system privilege resides at the bottom level in any role hierarchy.

For default Mode, parent_role_name column is NULL and role_level is 1.

Example

This example assumes these GRANT statements have been executed:

```
GRANT SERVER OPERATOR TO r4;
GRANT BACKUP DATABASE TO r3 WITH ADMIN OPTION;
GRANT DROP CONNECTION TO r3 WITH ADMIN ONLY OPTION;
GRANT MONITOR TO r2;GRANT CHECKPOINT TO r1;
GRANT ROLE r2 TO r1 WITH ADMIN OPTION;
GRANT ROLE r3 TO r2 WITH NO ADMIN OPTION;
GRANT ROLE r4 TO r3 WITH ADMIN ONLY OPTION;
GRANT ROLE r1 TO user1;
GRANT ROLE r1 TO r7;
GRANT ROLE r7 TO user2 WITH ADMIN OPTION;
GRANT BACKUP DATABASE TO user2 WITH ADMIN ONLY OPTION;
```

`sp_displayroles( 'user2', 'expand_down', 'ALL' )` produces output similar to:

role_name	parent_role_name	grant_type	role_level
r7	NULL	ADMIN	1
PUBLIC	NULL	NO ADMIN	1
BACKUP DATABASE	NULL	ADMIN ONLY	1
dbo	PUBLIC	NO ADMIN	2
r1	r7	NO ADMIN	2
r2	r1	ADMIN	3
CHECKPOINT	r1	NO ADMIN	3
r3	r2	NO ADMIN	4
MONITOR	r2	NO ADMIN	4
r4	r3	ADMIN ONLY	5
BACKUP DATABASE	r3	ADMIN	5
DROP CONNECTION	r3	ADMIN ONLY	5

`sp_displayroles( 'user2', 'expand_down', 'NO_ADMIN' )` produces output similar to:

role_name	parent_role_name	grant_type	role_level
r7	NULL	ADMIN	1
PUBLIC	NULL	NO ADMIN	1
dbo	PUBLIC	NO ADMIN	2
r1	r7	NO ADMIN	2
r2	r1	ADMIN	3
CHECKPOINT	r1	NO ADMIN	3
r3	r2	NO ADMIN	4
MONITOR	r2	NO ADMIN	4
BACKUP DATABASE	r3	ADMIN	5

`sp_displayroles( 'r3', 'expand_up', 'NO_ADMIN' )` produces out put similar to:

role_name	parent_role_name	grant_type	role_level
r1	r7	NO ADMIN	-2
r2	r1	ADMIN	-1
r3	r2	NO ADMIN	0

`sp_displayroles( 'r1', 'NO_ADMIN', 'expand_up')` produces output similar to:

role_name	parent_role_name	grant_type	role_level
r1	r7	NO ADMIN	0

4.3.4 sp_expireallpasswords System Procedure

Immediately expires all user passwords.

Syntax 1

```
call sp_expireallpasswords
```

Syntax 2

```
sp_expireallpasswords
```

Privileges

You must have EXECUTE privilege on the system procedure. You must also have the MANAGE ANY USER system privilege.

4.3.5 SP_HAS_ROLE Function [System]

Returns an integer value indicating whether the invoking user has been granted a specified system privilege or user-defined role. When used for privilege checking within user-defined stored procedures, `SP_HAS_ROLE` returns an error message when a user fails a privilege check.

Syntax

```
dbo.sp_has_role( [ <rolename> ],  [ <grant_type> ], [ <throw_error> ]  )
```

Parameters

Parameters	Description
rolename	The name of a system privilege or user-defined role.
grant_type	Valid values are: ADMIN and NO ADMIN. If NULL or not specified, NO ADMIN is used by default.
throw_error	Valid values are: • **1** – display error message specified if system privilege or user-defined role is not granted to invoking user. • **0** – (default) do not display error message if specified system privilege or user-defined role is not granted to invoking user.

Returns

Value	Description
1	System privilege or user-defined role is granted to invoking user.
0 or `Permission denied: you do not have permission to execute this command/procedure.`	System privilege or user-defined role is not granted to invoking user. The error message replaces the value 0 when the `throw_error` argument is set to 1.
-1	The system privilege or user-defined role specified does not exist. No error message appears, even if the `throw_error` argument is set to 1.

Remarks

If the value of the `grant_type` argument is ADMIN, the function checks whether the invoking user has administrative privileges for the system privilege. If the value of the `grant_type` argument is NO ADMIN, the function checks whether the invoking user has privileged use of the system privilege or role.

If the `grant_type` argument is not specified, NO ADMIN is used by default and output indicates only whether the invoking user has been granted, either directly or indirectly, the specified system privilege or user-defined role.

If the `rolename` and `grant_type` arguments are both NULL and the `throw_error` argument is 1, you see an error message. You may find this useful for those stored procedures where an error message appears after certain values are read from the catalog tables rather than after the checking the presence of certain system privileges for the invoking user.

> i Note
>
> A permission denied error message is returned if the arguments `rolename` and `grant_type` are set to NULL and `throw_error` is set to 1, or if all three arguments are set to NULL.

Example

The examples use the following scenario:

- `u1` has been granted the CREATE ANY PROCEDURE system privilege with the WITH NO ADMIN OPTION clause.
- `u1` has not been granted the CREATE ANY TABLE system privilege.
- `u1` has been granted the user-defined role `Role_A` with the WITH ADMIN ONLY OPTION clause.
- `Role_B` exists, but has not been granted to `u1`
- The role `Role_C` does not exist.

The examples are as follows:

- This example returns the value 1, which indicates `u1` has been granted the CREATE ANY PROCEDURE system privilege:

  ```
  sp_has_role 'create any procedure'
  ```

- This example returns the value 0, which indicates `u1` has not been granted the CREATE ANY TABLE system privilege. No error message is returned because the `throw_error` argument is not specified:

  ```
  sp_has_role 'create any table'
  ```

- This example returns the `Permission denied` error message (throw_error=1). Even though `u1` has been granted the CREATE ANY PROCEDURE system privilege, `u1` has not been granted administrative rights to the system privilege:

  ```
  sp_has_role 'create any procedure','admin',1
  ```

- This example returns the value 1, which indicates u1 has been granted role Role_A:

```
sp_has_role 'Role_A'
```

- This example returns the value 1, which indicates u1 has been granted role Role_A with administrative rights:

```
sp_has_role 'Role_A','admin',1
```

- This example returns the value 0, which indicates u1 has not been granted the role ROLE_B. No error message is returned because the throw_error argument is not specified:

```
sp_has_role 'Role_B'
```

- This example returns the value -1, which indicates the role ROLE_C does not exist:

```
sp_has_role 'Role_C'
```

- This example returns the value -1, which indicates the role ROLE_C does not exist:

```
sp_has_role 'Role_C',NULL,1
```

4.3.6 sp_iqaddlogin Procedure

Adds a new SAP IQ user account to the specified login policy.

> **i Note**
>
> Though sp_iqaddlogin is still supported for backwards compatibility, use CREATE USER to create new users.

Syntax 1

```
call sp_iqaddlogin (‘<username_in>’, ‘<pwd>’,
[ ’<password_expiry_on_next_login >’] [ , ’<policy_name >’] )
```

Syntax 2

```
sp_iqaddlogin ‘<username_in>’, ‘<pwd>’, [ ’<password_expiry_on_next_login >’]
[ , ’<policy_name >’]
```

Syntax 3

```
sp_iqaddlogin <username_in>, <pwd>, [ <password_expiry_on_next_login >] [ ,
<policy_name >]
```

Parameters

username_in The user's login name. Login names must conform to the rules for identifiers.
pwd The user's password. Passwords must conform to rules for passwords, that is, they must be valid identifiers.
password_expiry_on_next_login (Optional) Specifies whether user's password expires as soon as this user's login is created. Default setting is OFF (password does not expire).
policy_name (Optional) Creates the user under the named login policy. If unspecified, user is created under the root login policy.

A `<username_in/pwd>` created using `sp_iqaddlogin` and set to expire in one day is valid all day tomorrow and invalid on the following day. In other words, a login created today and set to expire in`< n>` days are not usable once the date changes to the `<(n+1)>`th day.

Privileges

You must have EXECUTE privilege on the system procedure. You must also have the MANAGE ANY USER system privilege.

Remarks

Adds a new SAP IQ user account, assigns a login policy to the user and adds the user to the ISYSUSER system table. If the user already has a user ID for the database but is not in `ISYSUSER`, (for example, if the user ID was added using the `GRANT CONNECT` statement or SAP IQ Cockpit), `sp_iqaddlogin` adds the user to the table.

If you do not specify a login policy name when calling the procedure, SAP IQ assigns the user to the root login policy.

> **i Note**
>
> If the maximum number of logins for a login policy is unlimited, then a user belonging to that login policy can have an unlimited number of connections.

The first user login forces a password change and assigns a login policy to the newly created user.

Example

These calls add the user rose with a password irk324 under the login policy named expired_password. This example assumes the expired_password login policy already exists.

```
call sp_iqaddlogin('rose', 'irk324', 'ON', 'expired_password')
```

```
sp_iqaddlogin 'rose','irk324', 'ON', 'expired_password'
```

4.3.7 sp_iqbackupdetails Procedure

Shows all the dbfiles included in a particular backup.

Syntax

```
sp_iqbackupdetails  <backup_id>
```

Parameters

backup_id Specifies the backup operation transaction identifier.

> i Note
>
> You can obtain the backup_id value from the SYSIQBACKUPHISTORY table by executing the query:
>
> ```
> select * from sysiqbackuphistory
> ```

Privileges

You must have EXECUTE privilege on the system procedure.

Remarks

sp_iqbackupdetails returns:

Column Name	Description
backup_id	Identifier for the backup transaction.
backup_time	Time of the backup.
backup_type	Type of backup: “Full,” “Incremental since incremental,” or “Incremental since full.”
selective_type	Subtype of backup: "All inclusive," "All RW files in RW dbspaces," "Set of RO dbspace/ file."
depends_on_id	Identifier for previous backup that the backup depends on.
dbspace_id	Identifier for the dbspace being backed up.
dbspace_name	Name of the dbspace from SYSIQBACKUPHISTORYDETAIL. If dbspace name matches the dbspace name in SYSDBSPACE for a given dbspace_id. Otherwise “null.”
dbspace_rwstatus	“ReadWrite” or “Read Only.”
dbspace_createid	Dbspace creation transaction identifier.
dbspace_alterid	Alter DBSPACE read-write mode transaction identifier.
dbspace_online	Status “Online” or “Offline.”
dbspace_size	Size of dbspace, in KB, at time of backup.
dbspace_backup_size	Size of data, in KB, backed up in the dbspace.
dbfile_id	Identifier for the dbfile being backed up.
dbfile_name	The logical file name, if it was not renamed after the backup operation. If renamed, “null.”
dbfile_rwstatus	“ReadWrite” or “Read Only.”
dbfile_createid	Dbfile creation transaction identifier.
dbfile_alterid	Alter DBSPACE alter FILE read-write mode transaction identifier
dbfile_size in MB	Size of the dbfile, in MB.
dbfile_backup_size	Size of the dbfile backup, in KB.
dbfile_path	The dbfile path from SYSBACKUPDETAIL, if it matches the physical file path (“file_name”) in SYSDBFILE for a given dbspace_id and the dbfile_id. Otherwise “null.”

Example

Sample output from sp_iqbackupdetails:

```
backup_id   backup_time             backup_type   selective_type   depends_on_id
      883   2008-09-23 13:58:49.0   Full          All inclusive                0
dbspace_id   dbspace_name   dbspace_rwstatus   dbspace_createid
         0   system         ReadWrite                         0
dbspace_alterid   dbspace_online dbspace_size dbspace_backup_size dbfile_id
              0                0         2884                2884         0
dbfile_name dbfile_rwstatus dbfile_createid dbfile_alterid dbfile_size
```

```
     system        ReadWrite                   0               0       2884
```

```
dbfile_backup_size dbfile_path
             2884  C:\\Documents and Settings\\All Users\\SybaseIQ\\demo\\iqdemo.db
```

4.3.8 sp_iqbackupsummary Procedure

Summarizes backup operations performed.

Syntax

```
sp_iqbackupsummary [ <timestamp or backup_id> ]
```

Parameters

timestamp or backup_id specifies the interval for which to report backup operations. If you specify a timestamp or a backup ID, only those records with backup_time greater than or equal to the time you enter are returned. If you specify no timestamp, the procedure returns all the backup records in ISYSIQBACKUPHISTORY.

Privileges

You must have EXECUTE privilege on the system procedure.

Remarks

Table 13: sp_iqbackupsummary Columns

Column Name	Description
backup_id	Identifier for the backup transaction
backup_time	Time of the backup
backup_type	Type of backup: “Full,” “Incremental since incremental,” “Incremental since full”, or "PITR"

Column Name	Description
selective_type	Subtype of backup: "All Inclusive," "All RW files in RW dbspaces," "Set of RO dbspace/ file"
virtual_type	Type of virtual backup: “Non-virtual,” “Decoupled,” or “Encapsulated”
depends_on_id	Identifier for backup that the backup depends on
creator	Creator of the backup
backup_size	Size, in KB, of the backup
user_comment	User comment
backup_command	The backup statement issued (minus the comment)

Example

Sample output of `sp_iqbackupsummary`:

```
backup_id   backup_time             backup_type   selective_type   virtual_type
      883   2008-09-23 13:58:49.0   Full          All inclusive    Non virtual
depends_on_id   creator   backup_size   user_comment   backup_command
            0   DBA                5864               backup database to
                                                         'c:\\\\temp\\\\b1'
```

4.3.9 sp_iqconnection Procedure

Shows information about connections and versions, including which users are using temporary dbspace, which users are keeping versions alive, what the connections are doing inside SAP IQ, connection status, database version status, and so on.

Syntax

```
sp_iqconnection [ <connhandle> ]
```

Applies to

Simplex and multiplex.

Privileges

You must have EXECUTE privilege on the system procedure. You must also have one of the following system privileges:

- DROP CONNECTION
- MONITOR
- SERVER OPERATOR

Remarks

`<connhandle>` is equal to the `Number` connection property and is the ID number of the connection. The `connection_property` system function returns the connection ID:

```
SELECT connection_property ( 'Number' )
```

When called with an input parameter of a valid `<connhandle>`, `sp_iqconnection` returns the one row for that connection only.

`sp_iqconnection` returns a row for each active connection. The columns ConnHandle, Name, Userid, LastReqTime, ReqType, CommLink, NodeAddr, and LastIdle are the connection properties Number, Name, Userid, LastReqTime, ReqType, CommLink, NodeAddr, and LastIdle respectively, and return the same values as the system function `sa_conn_info`. The additional columns return connection data from the SAP IQ side of the SAP IQ engine. Rows are ordered by ConnCreateTime.

The column MPXServerName stores information related to internode communication (INC), as shown:

Server Where Run	MPXServerName Column Content
Simplex server	NULL (All connections are local/user connections)
Multiplex coordinator	• NULL for local/user connections. • Contains value of secondary node's server name (source of connection) for every INC connection (either on-demand or dedicated heartbeat connection).
Multiplex secondary	• NULL for local/user connections. • Contains value of coordinator's server name (source of connection).

In Java applications, specify SAP IQ-specific connection properties from TDS clients in the RemotePWD field. This example, where `myconnection` becomes the IQ connection name, shows how to specify IQ specific connection parameters:

```
p.put("RemotePWD",",,CON=myconnection");
```

Column Name	Description
ConnHandle	The ID number of the connection
Name	The connection name specified by the ConnectionName (CON) connection parameter.

Column Name	Description
Userid	The user ID for the connection.
LastReqTime	The time at which the last request for the specified connection started.
ReqType	A string for the type of the last request.
IQCmdType	SAP IQ side, if any. The command type reflects commands defined at the implementation level of the engine. These commands consist of transaction commands, DDL and DML commands for data in the IQ store, internal IQ cursor commands, and special control commands such as The current command executing on the `OPEN` and `CLOSE`, `ConnHandle Name Userid LastReqTime ReqTypeThe current commandBACKUP DATABASE`, `RESTORE DATABASE`, and others.
LastIQCmdTime	The time the last IQ command started or completed on the IQ side of the SAP IQ engine on this connection.
IQCursors	The number of cursors open in the IQ store on this connection.
LowestIQCursorState	The IQ cursor state, if any. If multiple cursors exist on the connection, the state that appears is the lowest cursor state of all the cursors; that is, the furthest from completion. Cursor state reflects internal SAP IQ implementation detail and is subject to change in the future. Cursor states are: **NONE** No cursors have been rendered. **INITIALIZED** Cursor is initialized. **PARSED** Cursor is parsed and costed. **DESCRIBED** Cursor information, such as column information, has been put into the descriptor. **COSTED** Cursor cost has been costed. **PREPARED** Statement has been prepared; cursor is executing. **EXECUTED** After PREPARED, cursor state transitions to EXECUTED. **FETCHING** Fetching rows from cursor result set. **END_OF_DATA** Seen the last record. **CLOSED** Closed; DFO tree stays in place for re-open. **COMPLETED** Is completed. Tearing down DFO tree. **INVALID** Unrecoverable error occurred. As suggested by the names, the cursor state changes at the end of the operation. A state of PREPARED, for example, indicates that the cursor is executing.
IQthreads	The number of SAP IQ threads currently assigned to the connection. Some threads may be assigned but idle. This column can help you determine which connections are using the most resources.
TxnID	The transaction ID of the current transaction on the connection. This is the same as the transaction ID in the `.iqmsg` file by the BeginTxn, CmtTxn, and PostCmtTxn messages, as well as the Txn ID Seq logged when the database is opened.
ConnCreateTime	The time the connection was created.
TempTableSpaceKB	The number of kilobytes of IQ temporary store space in use by this connection for data stored in IQ temp tables.
TempWorkSpaceKB	The number of kilobytes of IQ temporary store space in use by this connection for working space such as sorts, hashes, and temporary bitmaps. Space used by bitmaps or other objects that are part of indexes on SAP IQ temporary tables are reflected in TempTableSpaceKB.

Column Name	Description
IQConnID	The ten-digit connection ID included as part of all messages in the `.iqmsg` file. This is a monotonically increasing integer unique within a server session.
satoiq_count	An internal counter used to display the number of crossings from the SAP SQL Anywhere side to the IQ side of the SAP IQ engine. This might be occasionally useful in determining connection activity. Result sets are returned in buffers of rows and do not increment satoiq_count or iqtosa_count once per row.
iqtosa_count	An internal counter used to display the number of crossings from the IQ side to the SAP SQL Anywhere side of the SAP IQ engine. This might be occasionally useful in determining connection activity.
CommLink	The communication link for the connection. This is one of the network protocols supported by SAP IQ, or is local for a same-machine connection.
NodeAddr	The node for the client in a client/server connection.
LastIdle	The number of ticks between requests.
MPXServerName	If an INC connection, the varchar(128) value contains the name of the multiplex server where the INC connection originates. NULL if not an INC connection.
LSName	The logical server name of the connection. NULL if logical server context is unknown or not applicable.
INCConnName	The name of the underlying INC connection for a user connection. The data type for this column is varchar(255). If `sp_iqconnection` shows an INC connection name for a suspended user connection, that user connection has an associated INC connection that is also suspended.
INCConnSuspended	The value "Y" in this column indicates that the underlying INC connection for a user connection is in a suspended state. The value "N" indicates that the connection is not suspended.

4.3.10 sp_iqcopyloginpolicy Procedure

Creates a new login policy by copying an existing one.

Syntax 1

```
call sp_iqcopyloginpolicy ('<existing-policy-name>', '<new-policy-name>' )
```

Syntax 2

```
sp_iqcopyloginpolicy '<existing-policy-name>', '<new-policy-name>'
```

Parameters

existing-policy-name The login policy to copy.
new-policy-name Name of the new login policy to create (CHAR(128)).

Privileges

You must have EXECUTE privilege on the system procedure. You must also have the MANAGE ANY LOGIN POLICY system privilege.

Example

Creates a new login policy named `<lockeduser>` by copying the login policy option values from the existing login policy named `<root>`:

```
call sp_iqcopyloginpolicy ('root','lockeduser')
```

4.3.11 sp_iqdbspace Procedure

Displays detailed information about each SAP IQ dbspace.

Syntax

```
sp_iqdbspace [ <dbspace-name> ]
```

Applies to

Simplex and multiplex.

Privileges

You must have EXECUTE privilege on the system procedure. You must also have the MANAGE ANY DBSPACE system privilege.

Remarks

Use the information from `sp_iqdbspace` to determine whether data must be moved, and for data that has been moved, whether the old versions have been deallocated.

Column Name	Description
DBSpaceName	Name of the dbspace as specified in the `CREATE DBSPACE` statement. Dbspace names are always case-insensitive, regardless of the `CREATE DATABASE...CASE IGNORE` or `CASE RESPECT` specification.
DBSpaceType	Type of the dbspace (`MAIN`, `SHARED_TEMP`, `TEMPORARY`, `RLV`, or `CACHE`).
Writable	T (writable) or F (not writable).
Online	T (online) or F (offline).
Usage	Percent of dbspace currently in use by all files in the dbspace.
TotalSize	Total size of all files in the dbspace in the units B (bytes), K (kilobytes), M (megabytes), G (gigabytes), T (terabytes), or P (petabytes).
Reserve	Total reserved space that can be added to all files in the dbspace.
NumFiles	Number of files in the dbspace.
NumRWFiles	Number of read-write files in the dbspace.
Stripingon	T (on), F (off).
StripeSize	Always 1, if disk striping is on.
BlkTypes	Space used by both user data and internal system structures.
OkToDrop	"Y" indicates the dbspace can be dropped; otherwise "N".
lsname	The logical server associated with the DAS dbspace.

Values of the BlkTypes block type identifiers:

Identifier	Block Type
A	Active version
B	Backup structures
C	Checkpoint log
D	Database identity
F	Free list
G	Global free list manager
H	Header blocks of the free list
I	Index advice storage
M	Multiplex CM*
O	Old version

Identifier	Block Type
R	RLV free list manager. The manager first reserves the blocks from the main store freelist and marks them as free. As RLV logging uses these blocks, they are marked as in use.
RC	Number of blocks actually in use by RLV store logs
RU	Number of blocks used by the commit log
T	Table use
U	Index use
N	Column use
X	Drop at checkpoint

*The multiplex commit identity block (actually 128 blocks) exists in all SAP IQ databases, even though it is not used by simplex databases.

Example

Displays information about dbspaces:

DBSpaceName	DBSpaceType	Writable	Online	Usage
iq_main	MAIN	T	T	26
IQ_SYSTEM_LOG	PITR	T	T	0
IQ_SYSTEM_MAIN	MAIN	T	T	22
IQ_SYSTEM_MAIN	TEMPORARY	T	T	23
rvspace	RLV	T	T	17

TotalSize	Reserve	NumRWFiles	NumFiles	Stripingon
100M	200M	1	1	T
0B	0B	1	1	F
100M	200M	1	1	T
25M	200M	1	1	T
1000M	0B	1	1	F

StripSize	BlkTypes	OkToDrop	lsname
1K	1H,3254A	N	(NULL)
0B	1H	N	(NULL)
1K	1H,2528F,32D,128M	N	(NULL)
1K	1H,64F,16A	N	(NULL)
1K	1H,20480R,2096RU,1040RC	N	lsname

Note that for the rvspace RLV dbsapce, in the BlkTypes column, of the 20480 blocks reserved for RLV store logs (20489R), 2096 blocks are in use (RU), 1040 blocks (RC) of which are in use by the commit log.

4.3.12 sp_iqdbspaceinfo Procedure

Displays the size of each object and subobject used in the specified table. Not supported for RLV dbspaces.

Syntax

```
sp_iqdbspaceinfo [ <dbspace-name> ] [ , <owner_name> ] [ ,
<object_name> ] [ , <object-type>  ]
```

Parameters

All parameters are optional, and any parameter may be supplied independent of another parameter's value.

dbspace_name if specified, sp_iqdbspaceinfo displays one line for each table that has any component in the specified dbspace. Otherwise, the procedure shows information for all dbspaces in the database.
owner_name owner of the object. If specified, sp_iqdbspaceinfo displays output only for tables with the specified owner. If not specified, sp_iqdbspaceinfo displays information on tables for all users in the database.
object_name name of the table. If not specified, sp_iqdbspaceinfo displays information on all tables in the database.
object_type valid table objects.

The sp_iqdbspaceinfo stored procedure supports wildcard characters for interpreting <dbspace_name>, <object_name>, and <owner_name>. It shows information for all dbspaces that match the given pattern in the same way the LIKE clause matches patterns inside queries.

Applies to

Simplex and multiplex.

Privileges

You must have EXECUTE privilege on the system procedure. You must also have one of the following system privileges:

- BACKUP DATABASE
- SERVER OPERATOR
- MANAGE ANY DBSPACE

Remarks

The procedure returns no results if you specify an RLV dbspace.

sp_iqdbspaceinfo shows the DBA the amount of space used by objects that reside on each dbspace. The DBA can use this information to determine which objects must be relocated before a dbspace can be dropped. The subobject columns display sizes reported in integer quantities followed by the suffix B, K, M, G, T, or P, representing bytes, kilobytes, megabytes, gigabytes, terabytes, and petabytes, respectively.

For tables, sp_iqdbspaceinfo displays subobject sizing information for all subobjects (using integer quantities with the suffix B, K, M, G, T, or P) sorted by <dbspace_name>, <object_name>, and <owner_name>.

Table 14: sp_iqdbspaceinfo Columns

Column Name	Description
dbspace_name	Name of the dbspace.
object_type	Type of the object (table or joinindex only).
owner	Name of the owner of the object.
object_name	Name of the object on the dbspace.
object_id	Global object ID of the object.
id	Table id of the object.
columns	Size of column storage space on the given dbspace.
indexes	Size of index storage space on the given dbspace. Does not use system-generated indexes (for example, HG indexes in unique constraints or FP indexes).
metadata	Size of storage space for metadata objects on the given dbspace.
primary_key	Size of storage space for primary key related objects on the given dbspace.
unique_constraint	Size of storage space for unique constraint-related objects on the given dbspace.
foreign_key	Size of storage space for foreign-key-related objects on the given dbspace.
dbspace_online	Indicates if the dbspace is online (Y) or offline (N).

If you run sp_iqdbspaceinfo against a server you have started with the -r switch (read-only), you see the error Msg 13768, Level 14, State 0: SAP SQL Anywhere Error -757: Modifications not permitted for read-only database. This behavior is expected. The error does not occur on other stored procedures such as sp_iqdbspace, sp_iqfile, sp_iqdbspaceobjectinfo or sp_iqobjectinfo.

Example

> **i Note**
>
> These examples show objects in the `iqdemo` database to better illustrate output. `iqdemo` includes a sample user dbspace named `iq_main` that may not be present in your own databases.

Displays the size of all objects and subobjects in all tables in all dbspaces in the database:

```
sp_iqdbspaceinfo
```

```
dbspace_name    object_type  owner   object_name     object_id  id   columns
iq_main         table        DBA     emp1              3689     741   96K
iq_main         table        DBA     iq_dummy          3686     740   24K
iq_main         table        DBA     sale              3698     742   96K
iq_main         table        GROUPO  Contacts          3538     732   288K
iq_main         table        GROUPO  Customers         3515     731   240K
iq_main         table        GROUPO  Departments       3632     738   72K
iq_main         table        GROUPO  Employees         3641     739   408K
iq_main         table        GROUPO  FinancialCodes    3612     736   72K
iq_main         table        GROUPO  FinancialData     3621     737   96K
iq_main         table        GROUPO  Products          3593     735   272K
iq_main         table        GROUPO  SalesOrderItems   3580     734   120K
iq_main         table        GROUPO  SalesOrders       3565     733   144K
indexes  metadata  primary_key  unique_constraint  foreign_key  dbspace_online
0B       1.37M     0B           0B                 0B           Y
0B       464K      0B           0B                 0B           Y
0B       1.22M     0B           0B                 0B           Y
0B       5.45M     24K          0B                 48K          Y
48K      4.63M     24K          0B                 0B           Y
0B       1.78M     24K          0B                 48K          Y
0B       8.03M     24K          0B                 48K          Y
0B       1.53M     24K          0B                 0B           Y
0B       2.19M     24K          0B                 48K          Y
192K     4.67M     24K          0B                 0B           Y
0B       2.7M      24K          0B                 104K         Y
0B       3.35M     24K          0B                 144K         Y
```

Displays the size of all objects and subobjects owned by a specified user in a specified dbspace in the database:

```
sp_iqdbspaceinfo iq_main,GROUPO
```

```
dbspace_name    object_type  owner   object_name     object_id  id   columns
iq_main         table        GROUPO  Contacts          3538     732   288K
iq_main         table        GROUPO  Customers         3515     731   240K
iq_main         table        GROUPO  Departments       3632     738   72K
iq_main         table        GROUPO  Employees         3641     739   408K
iq_main         table        GROUPO  FinancialCodes    3612     736   72K
iq_main         table        GROUPO  FinancialData     3621     737   96K
iq_main         table        GROUPO  Products          3593     735   272K
iq_main         table        GROUPO  SalesOrderItems   3580     734   120K
iq_main         table        GROUPO  SalesOrders       3565     733   144K
indexes  metadata  primary_key  unique_constraint  foreign_key  dbspace_online
0B       5.45M     24K          0B                 48K          Y
48K      4.63M     24K          0B                 0B           Y
0B       1.78M     24K          0B                 48K          Y
0B       8.03M     24K          0B                 48K          Y
0B       1.53M     24K          0B                 0B           Y
0B       2.19M     24K          0B                 48K          Y
192K     4.67M     24K          0B                 0B           Y
```

```
0B       2.7M      24K          0B                 104K         Y
0B       3.35M     24K          0B                 144K         Y
```

Displays the size of a specified object and its subobjects owned by a specified user in a specified dbspace in the database:

```
sp_iqdbspaceinfo iq_main,GROUPO,Departments
```

```
dbspace_name    object_type  owner   object_name     object_id  id   columns
iq_main         table        GROUPO  Departments       3632     738   72K
indexes  metadata  primary_key  unique_constraint  foreign_key  dbspace_online
0B       1.78M     24K          0B                 48K          Y
```

4.3.13 sp_iqdbspaceobjectinfo Procedure

Lists objects and subobjects of type table (including columns, indexes, metadata, primary keys, unique constraints, foreign keys, and partitions) for a given dbspace. Not supported for RLV dbspaces.

Syntax

```
sp_iqdbspaceobjectinfo [ <dbspace-name> ] [ , <owner_name> ] [ ,
<object_name> ] [ , <object-type> ]
```

Parameters

All parameters are optional and any parameter may be supplied independent of the value of other parameters.

dbspace-name If specified, sp_iqdbspaceobjectinfo displays output only for the specified dbspace. Otherwise, it shows information for all dbspaces in the database.

owner-name Owner of the object. If specified, sp_iqdbspaceobjectinfo displays output only for tables with the specified owner. If not specified, sp_iqdbspaceobjectinfo displays information for tables for all users in the database.

object-name Name of the table. If not specified, sp_iqdbspaceobjectinfo displays information for all tables in the database.

object-type Valid object types for table objects.

The sp_iqdbspaceobjectinfo stored procedure supports wildcard characters for interpreting <dbspace_name>, <object_name>, and <owner_name>. It displays information for all dbspaces that match the given pattern in the same way as the LIKE clause matches patterns inside queries.

Privileges

You must have EXECUTE privilege on the system procedure.

Remarks

The procedure returns no results if you specify an RLV dbspace.

For tables, `sp_iqdbspaceobjectinfo` displays summary information for all associated subobjects sorted by dbspace_name, owner and object_name.

`sp_iqdbspaceobjectinfo` displays the following information, based on the input parameter values:

Table 15: sp_iqdbspaceobjectinfo Columns

Column Name	Description
dbspace_name	Name of the dbspace.
dbspace_id	Identifier of the dbspace.
object_type	Table.
owner	Name of the owner of the object.
object_name	Name of the table object on the dbspace.
object_id	Global object ID of the object.
id	Table ID of the object.
columns	Number of table columns which are located on the given dbspace. If a column or one of the column-partitions is located on a dbspace, it is counted to be present on that dbspace. The result is shown in the form n/N (n out of total N columns of the table are on the given dbspace).
indexes	Number of user-defined indexes on the table which are located on the given dbspace. Shown in the form n/N (n out of total N indexes on the table are on the given dbspace). This does not contain indexes which are system-generated, such as FP indexes and HG indexes in the case of unique constraints.
metadata	Boolean field (Y/N) that denotes whether the metadata information of the subobject is also located on this dbspace.
primary_key	Boolean field (1/0) that denotes whether the primary key of the table, if any, is located on this dbspace.
unique_constraint	Number of unique constraints on the table that are located on the given dbspace. Appears in the form n/N (n out of total N unique constraints on the table are in the given dbspace).
foreign_key	Number of foreign_keys on the table that are located on the given dbspace. Appears in the form n/N (n out of total N foreign keys on the table are in the given dbspace).
partitions	Number of partitions of the table that are located on the given dbspace. Appears in the form n/N (n out of total N partitions of the table are in the given dbspace).

Example

These examples show objects in the `iqdemo` database to better illustrate output. `iqdemo` includes a sample user dbspace named `iq_main` that may not be present in your own databases.

Displays information about a specific dbspace in the database:

```
sp_iqdbspaceobjectinfo iq_main
```

```
dbspace_name  dbspace_id object_type  owner   object_name  object_id  id  columns
iq_main       16387      table        DBA     emp1           3689     741  4/4
iq_main       16387      table        DBA     iq_dummy       3686     740  1/1
iq_main       16387      table        DBA     sale           3698     742  4/4
iq_main       16387      table        GROUPO  Contacts       3538     732  12/12
iq_main       16387      table        GROUPO  Customers      3515     731  10/10
iq_main       16387      table        GROUPO  Departments    3632     738  3/3
iq_main       16387      table        GROUPO  Employees      3641     739  21/21
iq_main       16387      table        GROUPO  FinancialCodes 3612     736  3/3
iq_main       16387      table        GROUPO  FinancialData  3621     737  4/4
iq_main       16387      table        GROUPO  Products       3593     735  8/8
iq_main       16387      table        GROUPO  SalesOrderItems3580     734  5/5
iq_main       16387      table        GROUPO  SalesOrders    3565     733  6/6
indexes  metadata  primary_key  unique_constraint  foreign_key  partitions
0/0      Y                   0  0/0                0/0          0/0
0/0      Y                   0  0/0                0/0          0/0
0/0      Y                   0  0/0                0/0          0/0
0/0      Y                   1  0/0                1/1          0/0
1/1      Y                   1  0/0                0/0          0/0
0/0      Y                   1  0/0                1/1          0/0
0/0      Y                   1  0/0                1/1          0/0
0/0      Y                   1  0/0                0/0          0/0
0/0      Y                   1  0/0                1/1          0/0
4/4      Y                   1  0/0                0/0          0/0
0/0      Y                   1  0/0                2/2          0/0
0/0      Y                   1  0/0                3/3          0/0
```

Displays information about the objects owned by a specific user in a specific dbspace in the database:

```
sp_iqdbspaceobjectinfo iq_main,GROUPO
```

```
dbspace_name  dbspace_id object_type  owner   object_name    object_id  id  colum
ns
iq_main       16387      table        GROUPO  Contacts       3538       732 2/12
iq_main       16387      table        GROUPO  Customers      3515       731 10/10
iq_main       16387      table        GROUPO  Departments    3632       738 3/3
iq_main       16387      table        GROUPO  Employees      3641       739 21/21
iq_main       16387      table        GROUPO  FinancialCodes 3612       736 3/3
iq_main       16387      table        GROUPO  FinancialData  3621       737 4/4
iq_main       16387      table        GROUPO  Products       3593       735 8/8
iq_main       16387      table        GROUPO  SalesOrderItems3580       734 5/5
iq_main       16387      table        GROUPO  SalesOrders    3565       733 6/6
indexes  metadata  primary_key  unique_constraint  foreign_key  partitions
0/0      Y                   1  0/0                1/1          0/0
1/1      Y                   1  0/0                0/0          0/0
0/0      Y                   1  0/0                1/1          0/0
0/0      Y                   1  0/0                1/1          0/0
0/0      Y                   1  0/0                0/0          0/0
0/0      Y                   1  0/0                1/1          0/0
4/4      Y                   1  0/0                0/0          0/0
0/0      Y                   1  0/0                2/2          0/0
0/0      Y                   1  0/0                3/3          0/0
```

In this example, the commands move all tables on `dbspace_x` to `dbspace_y`.

```
SELECT 'ALTER TABLE ' || owner || '.' ||
object_name || ' MOVE TO dbspace_y;'
FROM sp_iqdbspaceobjectinfo()
WHERE object_type = 'table' AND
dbspace_name = 'dbspace_x';
```

The following `ALTER TABLE` commands are the result:

```
ALTER TABLE DBA.dt1 MOVE TO dbspace_y;
ALTER TABLE DBA.dt2 MOVE TO dbspace_y;
ALTER TABLE DBA.dt3 MOVE TO dbspace_y;
```

4.3.14 sp_iqdroplogin Procedure

Drops an SAP IQ user account.

Syntax 1

```
call sp_iqdroplogin (‘<userid>’)
```

Syntax 2

```
sp_iqdroplogin ‘<userid>’
```

Syntax 3

```
sp_iqdroplogin <userid>
```

Syntax 4

```
sp_iqdroplogin (‘<userid>’)
```

Parameters

userid ID of the user to drop.

Privileges

You must have EXECUTE privilege on the system procedure.

Remarks

`sp_iqdroplogin` drops the specified user.

Example

These commands all remove the user `rose`:

```
sp_iqdroplogin 'rose'
```

```
sp_iqdroplogin rose
```

```
call sp_iqdroplogin ('rose')
```

4.3.15 sp_iqemptyfile Procedure

Empties a dbfile and moves the objects in the dbfile to another available read-write dbfile in the same dbspace. Not available for files in an RLV dbspace.

Syntax

```
sp_iqemptyfile ( <logical-file--name> )
```

Privileges

You must have EXECUTE privilege on the system procedure. You must also have one of the following system privileges:

- BACKUP DATABASE
- SERVER OPERATOR
- ALTER DATABASE

In addition, you must also have one of the following system privileges:

- INSERT ANY TABLE
- UPDATE ANY TABLE
- DELETE ANY TABLE
- ALTER ANY TABLE
- LOAD ANY TABLE
- TRUNCATE ANY TABLE
- ALTER ANY OBJECT

Remarks

`sp_iqemptyfile` empties a dbfile. The dbspace must be read-write before you can execute the `sp_iqemptyfile` procedure. Dbfiles must be read-only before you can execute the `sp_iqemptyfile` procedure.. The procedure moves the objects in the file to another available read-write dbfile in the same dbspace. If there is no other read-write dbfile available, then SAP IQ displays an error message.

> ℹ Note
>
> In a shared multiplex environment, you can run `sp_iqemptyfile` only on the coordinator. There must be one read-write dbspace available for the procedure to succeed.

If the dbfile is in an RLV dbspace, then this error message displays:

```
Cannot empty files in an rlv store dbspace.
```

Example

Empties dbfile `das1`:

```
sp_iqemptyfile ('das1')
object_name
bytes_emptied          EmptiedFileSizePct
---------------------------------------
-------------          ------------------
admin_mpx.lineitem
180224                 0.0
admin_mpx.lineitem.ASIQ_IDX_T780_C10_FP
507904                 0.0
admin_mpx.lineitem.ASIQ_IDX_T780_C11_FP
5365760                0.0
```

```
admin_mpx.lineitem.ASIQ_IDX_T780_C12_FP
5488640              0.0
admin_mpx.lineitem.ASIQ_IDX_T780_C13_FP
5332992              0.0
admin_mpx.lineitem.ASIQ_IDX_T780_C14_FP
1048576              0.0
admin_mpx.lineitem.ASIQ_IDX_T780_C15_FP
1384448              0.0
admin_mpx.lineitem.ASIQ_IDX_T780_C16_FP
129589248            12.0
admin_mpx.lineitem.ASIQ_IDX_T780_C1_FP
6316032              0.0
admin_mpx.lineitem.ASIQ_IDX_T780_C2_FP
15523840             1.0
admin_mpx.lineitem.ASIQ_IDX_T780_C3_FP
7536640              0.0
admin_mpx.lineitem.ASIQ_IDX_T780_C4_FP
1056768              0.0
admin_mpx.lineitem.ASIQ_IDX_T780_C5_FP
2662400              0.0
admin_mpx.lineitem.ASIQ_IDX_T780_C6_FP
16949248             1.0
admin_mpx.lineitem.ASIQ_IDX_T780_C7_FP
1859584              0.0
admin_mpx.lineitem.ASIQ_IDX_T780_C8_FP
1966080              0.0
admin_mpx.lineitem.ASIQ_IDX_T780_C9_FP
843776               0.0
admin_mpx.lineitem.ASIQ_IDX_T780_I17_HG
29990912             2.0
admin_mpx.lineitem.big_l_orderkey_hg
30236672             2.0
admin_mpx.lineitem.big_l_partkey_hg
39034880             3.0
admin_mpx.lineitem.big_l_suppkey_hg
17547264             1.0
admin_mpx.orders
3473408              0.0
admin_mpx.orders.ASIQ_IDX_T781_C1_FP
12124160             1.0
admin_mpx.orders.ASIQ_IDX_T781_C2_FP
9175040              0.0
admin_mpx.orders.ASIQ_IDX_T781_C3_FP
892928               0.0
admin_mpx.orders.ASIQ_IDX_T781_C4_FP
21782528             2.0
admin_mpx.orders.ASIQ_IDX_T781_C5_FP
5496832              0.0
admin_mpx.orders.ASIQ_IDX_T781_C6_FP
1531904              0.0
admin_mpx.orders.ASIQ_IDX_T781_C7_FP
6307840              0.0
admin_mpx.orders.ASIQ_IDX_T781_C8_FP
262144               0.0
admin_mpx.orders.ASIQ_IDX_T781_C9_FP
239976448            22.0
```

4.3.16 sp_iqestdbspaces Procedure

Estimates the number and size of dbspaces needed for a given total index size.

Syntax

```
sp_iqestdbspaces ( <db_size_in_bytes>, <iq_page_size>,
<min_#_of_bytes>, <max_#_of_bytes> )
```

Privileges

You must have EXECUTE privilege on the system procedure. You must also have one of the following system privileges:

- MANAGE ANY DBSPACE
- ALTER DATABASE

Remarks

sp_iqestdbspaces reports several recommendations, depending on how much of the data is unique:

Recommendation	Description
min	If there is little variation in data, you can choose to create only the dbspace segments of the sizes recommended as min. These recommendations reflect the best possible compression on data with the least possible variation.
avg	If your data has an average amount of variation, create the dbspace segments recommended as min, plus additional segments of the sizes recommended as avg.
max	If your data has a high degree of variation (many unique values), create the dbspace segments recommended as min, avg, and max.
spare	If you are uncertain about the number of unique values in your data, create the dbspace segments recommended as min, avg, max, and spare. You can always delete unused segments after loading your data, but creating too few can cost you some time.

Displays information about the number and size of dbspace segments based on the size of the database, the IQ page size, and the range of bytes per dbspace segment. This procedure assumes that the database was created with the default block size for the specified IQ page size; otherwise, the returned estimated values are incorrect.

Table 16: sp_iqestdbspaces Parameters

Name	Datatype	Description
db_size_in_bytes	decimal(16)	Size of the database in bytes.
iq_page_size	smallint	The page size defined for the IQ segment of the database (must be a power of 2 between 65536 and 524288; the default is 131072).
min_#_of_bytes	int	The minimum number of bytes per dbspace segment. The default is 20,000,000 (20 MB).
max_#_of_bytes	int	The maximum number of bytes per dbspace segment. The default is 2,146,304,000 (2.146 GB).

4.3.17 sp_iqfile Procedure

Displays detailed information about each dbfile in a dbspace.

Syntax

```
sp_iqfile [ <dbspace-name> ]
```

Applies to

Simplex and multiplex.

Privileges

You must have EXECUTE privilege on the system procedure. You must also have the MANAGE ANY DBSPACE system privilege.

Remarks

sp_iqfile displays the usage, properties, and types of data in each dbfile in a dbspace. You can use this information to determine whether data must be moved, and for data that has been moved, whether the old versions have been deallocated.

Column Name	Description
DBSpaceName	Name of the dbspace as specified in the `CREATE DBSPACE` statement. Dbspace names are always case-insensitive, regardless of the `CREATE DATABASE`...`CASE IGNORE` or `CASE RESPECT` specification.
DBFileName	Logical file name.
Path	Location of the physical file or raw partition.
SegmentType	Type of dbspace (`MAIN`, `TEMPORARY`, `RLV`, or `CACHE`).
RWMode	Mode of the dbspace: always read-write (RW).
Online	T (online) or F (offline). T = both the online value of the file's associated dbspace and the online value of the file in SYS.ISYSIQDBFILE are T; otherwise F.
Usage	Percent of dbspace currently in use by this file in the dbspace. When run against a secondary node in a multiplex configuration, this column displays NA.
DBFileSize	Current size of the file or raw partition. For a raw partition, this size value can be less than the physical size.
Reserve	Reserved space that can be added to this file in the dbspace.
StripeSize	Always 1, if disk striping is on.
BlkTypes	Space used by both user data and internal system structures.
FirstBlk	First IQ block number assigned to the file.
LastBlk	Last IQ block number assigned to the file.
OkToDrop	The server that created the DAS dbfile.
MirrorLogicalFileName	Logical filename of the primary DAS dbfile.
IsDASSharedFile	"T" indicates that the DAS dbfile is a shared file system file; otherwise "F".

Identifier	Block Type
A	Active Version
B	Backup Structures
C	Checkpoint Log
D	Database Identity
F	Free List
G	Global Free List Manager

Identifier	Block Type
H	Header Blocks of the Free List
I	Index Advice Storage
M	Multiplex CM*
O	Old Version
R	RLV Free List manager
T	Table Use
U	Index Use
N	Column Use
X	Drop at Checkpoint

*The multiplex commit identity block (actually 128 blocks) exists in all SAP IQ databases, even though it is not used by simplex databases.

Example

Displays information about the files in the dbspaces:

```
sp_iqfile;
```

```
sp_iqfile;
DBSpaceName,DBFileName,Path,SegmentType,RWMode,Online,
Usage,DBFileSize,Reserve,StripeSize,BlkTypes,FirstBlk,
LastBlk,OkToDrop,servername,mirrorLogicalFileName,IsDASSharedFile
'IQ_SYSTEM_MAIN','IQ_SYSTEM_MAIN','../
mpx_configdb.iq','MAIN','RW','T','24','700M','0B','1K','1H,17888F,32D,2498A,151O,
198X,128M,32C',1,89600,'N',,'(NULL)','F'
'dbsp1','dbsp1','/lint12dev7/users/user4/machine.lint12dev_local/mpxstore/
mpx_configdb.iqdb1','MAIN','RW','T','1','50M','0B','1K','1H',
1045440,1051839,'N',,'(NULL)','F'
'dbsp2','dbsp2','/lint12dev7/users/user4/machine.lint12dev_local/mpxstore/
mpx_configdb.iqdb2','MAIN','RW','T','1','50M','0B','1K','1H',
2090880,2097279,'N',,'(NULL)','F'
'dbsp3','dbsp3','/lint12dev7/users/user4/machine.lint12dev_local/mpxstore/
mpx_configdb.iqdb3','MAIN','RW','T','1','50M','0B','1K','1H',
3136320,3142719,'N',,'(NULL)','F'
'dbsp4','dbsp4','/lint12dev7/users/user4/machine.lint12dev_local/mpxstore/
mpx_configdb.iqdb4','MAIN','RW','T','1','50M','0B','1K','1H',
4181760,4188159,'N',,'(NULL)','F'
'dbsp5','dbsp5','/lint12dev7/users/user4/machine.lint12dev_local/mpxstore/
mpx_configdb.iqdb5','MAIN','RW','T','1','50M','0B','1K','1H',
5227200,5233599,'N',,'(NULL)','F'
'dbsp6','dbsp6','/lint12dev7/users/user4/machine.lint12dev_local/mpxstore/
mpx_configdb.iqdb6','MAIN','RW','T','1','50M','0B','1K','1H',
6272640,6279039,'N',,'(NULL)','F'
```

```
'dbsp7','dbsp71','/lint12dev7/users/user4/machine.lint12dev_local/mpxstore/
mpx_configdb.iqdb71','MAIN','RW','T','1','200M','0B','1K','1H',
7318080,7343679,'Y',,'(NULL)','F'
'dbsp7','dbsp72','/lint12dev7/users/user4/machine.lint12dev_local/mpxstore/
mpx_configdb.iqdb72','MAIN','RW','T','1','200M','0B','1K','1H',
8363520,8389119,'Y',,'(NULL)','F'
'dbsp7','dbsp73','/lint12dev7/users/user4/machine.lint12dev_local/mpxstore/
mpx_configdb.iqdb73','MAIN','RW','T','1','200M','0B','1K','1H',
9408960,9434559,'Y',,'(NULL)','F'
'dbsp8','dbsp81','/lint12dev7/users/user4/machine.lint12dev_local/mpxstore/
mpx_configdb.iqdb81','MAIN','RW','T','1','20M','0B','1K','1H',
10454400,10456959,'Y',,'(NULL)','F'
'dbsp8','dbsp82','/lint12dev7/users/user4/machine.lint12dev_local/mpxstore/
mpx_configdb.iqdb82','MAIN','RW','T','1','20M','0B','1K','1H',
11499840,11502399,'Y',,'(NULL)','F'
'dbsp8','dbsp83','/lint12dev7/users/user4/machine.lint12dev_local/mpxstore/
mpx_configdb.iqdb83','MAIN','RW','T','1','20M','0B','1K','1H',
12545280,12547839,'Y',,'(NULL)','F'
'das62H2','dasP62H2','/lint12dev7/users/user4/machine.lint12dev_local/mpxstore/
das62H2_1.iq','MAIN','RW','T','1','10M','0B','1K','1H',
13590720,13591999,'Y','user4_1927_nw45780','(NULL)','F'
'das62H2','dasM62H2','/lint12dev7/users/user4/machine.lint12dev_local/mpxstore/
das62H2_3.iq','MAIN','RW','T','1','10M','0B','1K','1H',
14636160,14637439,'Y','user4_1927_nw55880','dasP62H2','F'
'das62H2','dasM62H2_11','/lint12dev7/users/user4/machine.lint12dev_local/
mpxstore/das62H2_33.iq','MAIN','RW','T','1','10M','0B','1K','1H',
15681600,15682879,'Y','user4_1927_nw45780','dasP62H2','F'
'das62H2','dasM62H2_22','/lint12dev7/users/user4/machine.lint12dev_local/
mpxstore/das62H2_44.iq','MAIN','RW','T','1','10M','0B','1K','1H',
16727040,16728319,'Y','user4_1927_nw45780','dasP62H2','F'
'das62H2','dasP62H_55','/lint12dev7/users/user4/machine.lint12dev_local/mpxstore/
das62H_55.iq','MAIN','RW','T','1','50M','0B','1K','1H',
17772480,17778879,'Y','user4_1927_nw55880','(NULL)','F'
'das62H2','dasM62H_55','/lint12dev7/users/user4/machine.lint12dev_local/mpxstore/
das62M_55.iq','MAIN','RW','T','1','50M','0B','1K','1H',
18817920,18824319,'Y','user4_1927_nw45780','dasP62H_55','F'
'sfs_dbs','f1','/shared_disk1/users/user4/dasfmpx/nw35095/
f1.iq','MAIN','RW','T','1','7.81M','0B','1K','1H',
2090880,2091879,'Y','nw35095_dbsrv7915','(NULL)','T'
'sfs_dbs','f1_m','/local_disk1/users/user4/dasfmpx/nw411359/
f1_m.iq','MAIN','RW','T','1','7.81M','0B','1K','1H',
2090880,2091879,'Y','nw411359_dbsrv7915','f1','F'
'sfs_dbs','f2','/shared_disk2/users/user4/dasfmpx/nw35095/
f2.iq','MAIN','RW','T','1','7.81M','0B','1K','1H',
3136320,3137319,'Y','nw35095_dbsrv7915','(NULL)','T'
'sfs_dbs','f2_m','/local_disk2/users/user4/dasfmpx/nw411359/
f2_m.iq','MAIN','RW','T','1','7.81M','0B','1K','1H',
3136320,3137319,'Y','nw411359_dbsrv7915','f2','F'
'IQ_SYSTEM_TEMP','IQ_SYSTEM_TEMP','nc110203mpx_configdb.iqtmp','TEMPORARY','RW','
T','1','300M','0B','1K','1H,64F,48A',
1,38400,'N','tbucken_1927_nc110203','(NULL)','F'
```

4.3.18 sp_iqmodifyadmin Procedure

Sets an option on a named login policy to a certain value. If no login policy is specified, the option is set on the root policy. In a multiplex, `sp_iqmodifyadmin` takes an optional parameter that is the multiplex server name.

Syntax1

```
call sp_iqmodifyadmin ('<policy_option_name>', '<value_in>' ,
['<login_policy_name>'] )
```

Syntax2

```
sp_iqmodifyadmin '<policy_option_name>', '<value_in'> ,'<login_policy_name >'
```

Syntax3

```
sp_iqmodifyadmin <policy_option_name>, <value_in>, ,<login_policy_name>
```

Syntax 4

```
sp_iqmodifyadmin '<policy_option_name>',
'<value_in'> ,'<login_policy_name >' ,'<server_name >'
```

Parameters

policy_option_name The login policy option to be changed.
value_in New value for the login policy option.
login_policy_name Policy for which the login policy option is to be changed.

Privileges

You must have EXECUTE privilege on the system procedure. You must also have the MANAGE ANY LOGIN POLICY system privilege.

Example

Sets the login option `locked` to ON for the policy named `<lockeduser>`:

```
call sp_iqmodifyadmin ('locked','on','lockeduser')
```

Sets the login option `locked` to ON for the policy named `<lockeduser>` on the multiplex server named `Writer1`:

```
call sp_iqmodifyadmin ('locked','on','lockeduser','Writer1')
```

4.3.19 sp_iqmodifylogin Procedure

Assigns a user to a login policy.

Syntax 1

```
call sp_iqmodifylogin '<userid>', ['<login_policy_name>']
```

Syntax 2

```
sp_iqmodifylogin '<userid>', ['<login_policy_name>']
```

Parameters

userid Variable that holds the name of the account to modify.

login_policy_name (Optional) Specifies the name of the login policy to which the user will be assigned. If no login policy name is specified, the user is assigned to the root login policy.

Privileges

You must have EXECUTE privilege on the system procedure. You must also have the MANAGE ANY USER system privilege.

Example

Assigns user joe to a login policy named expired_password:

```
sp_iqmodifylogin 'joe', 'expired_password'
```

Assigns user joe to the root login policy:

```
call sp_iqmodifylogin ('joe')
```

4.3.20 sp_iqobjectinfo Procedure

Returns partitions and dbspace assignments of database objects and subobjects.

Syntax

```
sp_iqobjectinfo [ <owner_name> ]  [ , <object_name> ] [ , <object-type> ]
```

Parameter

owner_name Owner of the object. If specified, sp_iqobjectinfo displays output only for tables with the specified owner. If not specified, sp_iqobjectinfo displays information on tables for all users in the database.
object_name Name of the table. If not specified, sp_iqobjectinfo displays information on all tables in the database.
object-type

Valid table object types.

If the object-type is a table, it must be enclosed in quotation marks.

All parameters are optional, and any parameter may be supplied independent of the value of another parameter.

Privileges

You must have EXECUTE privilege on the system procedure.

Remarks

Use input parameters with `sp_iqobjectinfo`; you can query the results of the `sp_iqobjectinfo` and it performs better if you use input parameters rather than using predicates in the `WHERE` clause of the query. For example, Query A is written as:

```
SELECT COUNT(*) FROM sp_iqobjectinfo()
WHERE owner = 'DBA'
AND object_name = 'tab_case510'
AND object_type = 'table'
AND sub_object_name is NULL
AND dbspace_name = 'iqmain7'
AND partition_name = 'P1'
```

Query B is Query A rewritten to use `sp_iqobjectinfo` input parameters:

```
SELECT COUNT(*) FROM sp_iqobjectinfo('DBA','tab_case510','table')
WHERE sub_object_name is NULL
AND dbspace_name = 'iqmain7'
AND PARTITION_NAME = 'P1'
```

Query B returns results faster than Query A. When the input parameters are passed to `sp_iqobjectinfo`, the procedure compares and joins fewer records in the system tables, thus doing less work compared to Query A. In Query B, the predicates are applied in the procedure itself, which returns a smaller result set, so a smaller number of predicates is applied in the query.

The `sp_iqobjectinfo` stored procedure supports wildcard characters for interpreting `<owner_name>`, `<object_name>`, and `<object_type>`. It shows information for all dbspaces that match the given pattern in the same way the `LIKE` clause matches patterns inside queries.

Returns all the partitions and the dbspace assignments of a particular or all database objects (of type table) and its subobjects. The subobjects are columns, indexes, primary key, unique constraints, and foreign keys.

Table 17: sp_iqobjectinfo Columns

Column Name	Description
owner	Name of the owner of the object.
object_name	Name of the object (of type table) located on the dbspace.
sub_object_name	Name of the object located on the dbspace.
object_type	Type of the object (column, index, primary key, unique constraint, foreign key, partition, or table).
object_id	Global object ID of the object.
id	Table ID of the object.

Column Name	Description
dbspace_name	Name of the dbspace on which the object resides. The string “[multiple]” appears in a special meta row for partitioned objects. The [multiple] row indicates that multiple rows follow in the output to describe the table or column.
partition_name	Name of the partition for the given object.

Example

i Note

These examples show objects in the `iqdemo` database to better illustrate output. `iqdemo` includes a sample user dbspace named `iq_main` that may not be present in your own databases.

Displays information about partitions and dbspace assignments of a specific database object and subobjects owned by a specific user:

```
sp_iqobjectinfo GROUPO,Departments
```

```
owner   object_name  sub_object_name                object_type  object_id  id
GROUPO  Departments  (NULL)                          table          3632    738
GROUPO  Departments  DepartmentID                    column         3633    738
GROUPO  Departments  DepartmentName                  column         3634    738
GROUPO  Departments  DepartmentHeadID                column         3635    738
GROUPO  Departments  DepartmentsKey                  primary key    83      738
GROUPO  Departments  FK_DepartmentHeadID_EmployeeID  foreign key    92      738
dbspace_name     partition_name
iq_main          (NULL)
iq_main          (NULL)
iq_main          (NULL)
iq_main          (NULL)
iq_main          (NULL)
iq_main          (NULL)
```

Displays information about partitions and dbspace assignments of a specific database object and subobjects owned by a specific user for `<object-type>` `table`:

```
sp_iqobjectinfo DBA,sale,'table'
```

```
owner  object_name  sub_object_name  object_type  object_id  id
DBA    sale          (NULL)           table         3698     742
DBA    sale         prod_id           column        3699     742
DBA    sale         month_num         column        3700     742
DBA    sale         rep_id            column        3701     742
DBA    sale         sales             column        3702     742
dbspace_name      partition_name
iq_main           (NULL)
iq_main           (NULL)
iq_main           (NULL)
iq_main           (NULL)
iq_main           (NULL)
```

4.3.21 sp_iqspaceused Procedure

Shows information about space available and space used in the IQ store, IQ temporary store, RLV store, and IQ global and local shared temporary stores.

Syntax

```
sp_iqspaceused(out mainKB            unsigned bigint,
               out mainKBUsed        unsigned bigint,
               out tempKB            unsigned bigint,
               out tempKBUsed        unsigned bigint,
               out shTempTotalKB     unsigned bigint,
               out shTempTotalKBUsed unsigned bigint,
               out shTempLocalKB     unsigned bigint,
               out shTempLocalKBUsed unsigned bigint,
               out rlvLogKB          unsigned bigint,
               out rlvLogKBUsed      unsigned bigint)
```

Applies to

Simplex and multiplex.

Privileges

You must have EXECUTE privilege on the system procedure. You must also have one of the following system privilege:

- ALTER DATABASE
- MANAGE ANY DBSPACE
- MONITOR

Remarks

`sp_iqspaceused` returns several values as unsigned bigint out parameters. This system stored procedure can be called by user-defined stored procedures to determine the amount of main, temporary, and RLV store space in use.

`sp_iqspaceused` returns a subset of the information provided by `sp_iqstatus`, but allows the user to return the information in SQL variables to be used in calculations.

If run on a multiplex database, this procedure applies to the server on which it runs. Also returns space used on IQ_SHARED_TEMP.

Column Name	Description
mainKB	The total IQ main store space, in kilobytes.
mainKBUsed	The number of kilobytes of IQ main store space used by the database. Secondary multiplex nodes return '(Null)'.
tempKB	The total IQ temporary store space, in kilobytes.
tempKBUsed	The number of kilobytes of total IQ temporary store space in use by the database.
shTempTotalKB	The total IQ global shared temporary store space, in kilobytes.
shTempLocalKB	The total IQ local shared temporary store space, in kilobytes.
shTempLocalKBUsed	The number of kilobytes of IQ local shared temporary store space in use by the database.
rlvLogKB	The total RLV store space, in kilobytes.
rlvLogKBUsed	The number of kilobytes of RLV store space in use by the database.

Example

`sp_iqspaceused` requires seven output parameters. Create a user-defined stored procedure `myspace` that declares the seven output parameters, then calls `sp_iqspaceused`:

```
  create or replace procedure dbo.myspace()
begin
     declare mt unsigned bigint;
     declare mu unsigned bigint;
     declare tt unsigned bigint;
     declare tu unsigned bigint;
     declare gt unsigned bigint;
     declare gu unsigned bigint;
     declare lt unsigned bigint;
     declare lu unsigned bigint;
     declare tt_t unsigned bigint;
     declare mt_t unsigned bigint;
     declare gt_t unsigned bigint;
     declare lt_t unsigned bigint;
     call sp_iqspaceused(mt,mu,tt,tu,gt,gu,lt,lu);
     if (tt = 0) then
        set tt_t = 0;
     else
        set tt_t = tu*100/tt;
     end if;
     if (mt = 0) then
        set mt_t = 0;
     else
        set mt_t = mu*100/mt;
     end if;
     if (gt = 0) then
        set gt_t = 0;
     else
        set gt_t = gu*100/gt;
     end if;
     if (lt = 0) then
        set lt_t = 0;
     else
        set lt_t = lu*100/lt;
     end if;
  select cast(mt/1024 as unsigned bigint) as mainMB,
         cast(mu/1024 as unsigned bigint) as mainusedMB, mt_t as mainPerCent,
         cast(tt/1024 as unsigned bigint) as tempMB,
         cast(tu/1024 as unsigned bigint) as tempusedMB, tt_t as tempPerCent,
         cast(gt/1024 as unsigned bigint) as shTempTotalKB,
         cast(gu/1024 as unsigned bigint) as shTempTotalKBUsed, gt_t as
globalshtempPerCent,
         cast(lt/1024 as unsigned bigint) as shTempLocalMB,
         cast(lu/1024 as unsigned bigint) as shTempLocalKBUsed, lt_t as
localshtempPerCent;
end
```

To display the output of `sp_iqspaceused`, execute `myspace`:

```
myspace
```

4.3.22 sp_iqsysmon Procedure

Monitors multiple components of SAP IQ, including the management of buffer cache, memory, threads, locks, I/O functions, and CPU utilization.

Batch Mode Syntax

```
sp_iqsysmon start_monitor
sp_iqsysmon stop_monitor [, 'section(s)' ]
or
sp_iqsysmon '<time-period>' [, 'section(s)' ]
```

File Mode Syntax

```
sp_iqsysmon start_monitor, 'filemode' [, '<monitor-options>' ]
sp_iqsysmon stop_monitor
```

Batch Mode Parameters

start_monitor Starts monitoring.
stop_monitor Stops monitoring and displays the report.
time-period The time period for monitoring, in the form HH:MM:SS.
section(s)

The abbreviation for one or more sections to be shown by `sp_iqsysmon`.

See the Remarks [page 420] section for a complete list of abbreviations.

If you specify more than one section, separate the section abbreviations using spaces, and enclose the list in single or double quotes. The default is to display all sections.

For sections related to the IQ main store, you can specify main or temporary store by prefixing the section abbreviation with 'm' or 't', respectively. Without the prefix, both stores are monitored. For example, if you specify 'mbufman', only the IQ main store buffer manager is monitored. If you specify 'mbufman tbufman' or 'bufman', both the main and temporary store buffer managers are monitored.

> ℹ Note
>
> The SAP IQ components Disk I/O and Lock Manager are not currently supported by `sp_iqsysmon`.

File Mode Parameters

start_monitor Starts monitoring.
stop_monitor Stops monitoring and writes the remaining output to the log file.
filemode Specifies that `sp_iqsysmon` is running in file mode. In file mode, a sample of statistics appear for every interval in the monitoring period. By default, the output is written to a log file named `<dbname.connid-iqmon>`. Use the `file_suffix` option to change the suffix of the output file. See the `<monitor_options>` parameter for a description of the `file_suffix` option.
monitor_options The monitor _options string can include one or more options:

-interval seconds Specifies the reporting interval, in seconds. A sample of monitor statistics is output to the log file after every interval. The default is every 60 seconds, if the -interval option is not specified. The minimum reporting interval is 2 seconds. If the interval specified for this option is invalid or less than 2 seconds, the interval is set to 2 seconds.

The first display shows the counters from the start of the server. Subsequent displays show the difference from the previous display. You can usually obtain useful results by running the monitor at the default interval of 60 seconds during a query with performance problems or during a time of day that generally has performance problems. A very short interval may not provide meaningful results. The interval should be proportional to the job time; 60 seconds is usually more than enough time.
-file_suffix suffix Creates a monitor output file named `dbname.connid-suffix`. If you do not specify the -file_suffix option, the suffix defaults to `iqmon`. If you specify the -file_suffix option and do not provide a suffix or provide a blank string as a suffix, no suffix is used.
-append or -truncate Directs `sp_iqsysmon` to append to the existing output file or truncate the existing output file, respectively. Truncate is the default. If both options are specified, the option specified later in the string takes precedence.
-section section(s)

Specifies the abbreviation of one or more sections to write to the monitor log file.

See the Remarks [page 420] section for a complete list of abbreviations.

The default is to write all sections. The abbreviations specified in the sections list in file mode are the same abbreviations used in batch mode. When more than one section is specified, spaces must separate the section abbreviations.

If the -section option is specified with no sections, none of the sections are monitored. An invalid section abbreviation is ignored and a warning is written to the IQ message file.

Privileges

You must have EXECUTE privilege on the system procedure. You must also have the MONITOR system privilege.

Remarks

Report Sections or IQ Components to be Reported On	Abbreviation to Type
Buffer allocation	(main) – mbufalloc (temporary) – tbufalloc
Buffer manager	(main) – mbufman (temporary) – tbufman
Buffer pool	(main) – mbufpool (temporary) – tbufpool
Catalog statistics	catalog
CPU utilization	cpu
Free list management	(main)– mfreelist (temporary) – tfreelist
Memory management	memory
Prefetch management	(main)– mprefetch (temporary)– tprefetch
IQ RLV In-Memory Store statistics	rlv
Large Memory Allocator (LMA) statistics	lma
Server context statistics	server
Thread management	threads
Transaction management	txn

The `sp_iqsysmon` stored procedure monitors multiple components of SAP IQ, including the management of buffer cache, memory, threads, locks, I/O functions, and CPU utilization.

The `sp_iqsysmon` procedure supports two modes of monitoring:

Batch mode sp_iqsysmon collects the monitor statistics for the period between starting and stopping the monitor or for the time period specified in the <time-period> parameter. At the end of the monitoring period, sp_iqsysmon displays a list of consolidated statistics.

sp_iqsysmon in batch mode is similar to the SAP Adaptive Server Enterprise (SAP ASE) procedure sp_sysmon.

File mode

sp_iqsysmon writes the sample statistics in a log file for every interval period between starting and stopping the monitor.

The first display in file mode shows the counters from the start of the server. Subsequent displays show the difference from the previous display.

sp_iqsysmon in file mode is similar to the IQ UTILITIES command START MONITOR and STOP MONITOR interface.

Large Memory Allocator (LMA) Statistics

Definitions for the STATS-NAME abbreviations displayed in sp_iqsysmon output for LMA.

STATS-NAME	Definition
Large Memory Space	Maximum Large Memory configured size (-iqlm value from params.cfg).
Large Memory Max Flexible	Maximum memory granted for flexible operators. Example: Load Engine (hash sort merge for hash or hash-range partitioned table and hash sort merge cursor).
Large Memory Num Flex Allocations	This is the count of memory chunks allocated as flex memory.
Large Memory Flexible %	Percentage of large memory used for flexible operators.
Large Memory Flexible used	This is the total amount of memory allocated to flex users.
Large Memory Inflexible %	Percentage of large memory used for inflexible operators (N-bit metadata structures, data buffer of column vector in load).
Large Memory Inflexible used	Large memory used by inflexible operators.
Large Memory Anti-Starvation %	This only applies to flexible operators.
Large Memory Num Connections	(Internal use only)

Batch Mode Syntax Example

Example 1:

Starts the monitor in batch mode and displays all sections for the main and temporary stores:

```
sp_iqsysmon start_monitor
sp_iqsysmon stop_monitor
```

Example 2:

Starts the monitor in batch mode and displays the Buffer Manager and Buffer Pool statistics for the main store:

```
sp_iqsysmon start_monitor
sp_iqsysmon stop_monitor 'mbufman mbufpool'
```

Example 3:

Prints monitor information after 10 minutes:

```
sp_iqsysmon '00:10:00'
```

Example 4:

Prints only the Memory Manager section of the `sp_iqsysmon` report after 5 minutes:

```
sp_iqsysmon '00:05:00', memory
```

Example 5:

Starts the monitor, executes two procedures and a query, stops the monitor, then prints only the Buffer Manager section of the report:

```
sp_iqsysmon start_monitor
     go
     execute proc1
     go
     execute proc2
     go
     select sum(total_sales) from titles
     go
     sp_iqsysmon stop_monitor, bufman
     go
```

Example 6:

Prints only the Main Buffer Manager and Main Buffer Pool sections of the report after 2 minutes:

```
sp_iqsysmon '00:02:00', 'mbufman mbufpool'
```

Example 7:

Prints only the RLV sections of the report after 1 hour:

```
sp_iqsysmon '01:00:00','rlv'
```

Example 8:

Prints only the LMA sections of the report after 5 seconds:

```
sp_iqsysmon '00:00:05', 'lma'
```

Example 9:

Runs the monitor in batch mode for 10 seconds and displays the consolidated statistics at the end of the time period:

```
sp_iqsysmon '00:00:10', 'mbufpool memory'
```

File Mode Syntax Example

Example 1:

Truncates and writes information to the log file every 2 seconds between starting the monitor and stopping the monitor:

```
sp_iqsysmon start_monitor, 'filemode', '-interval 2'
     .
     .
     .
     sp_iqsysmon stop_monitor
```

Example 2:

Appends output for only the Main Buffer Manager and Memory Manager sections to an ASCII file with the name `dbname.connid-testmon`. For the database `iqdemo`, writes results in the file `iqdemo.2-testmon`:

```
sp_iqsysmon start_monitor, 'filemode',
     '-file_suffix testmon -append -section mbufman memory'
     .
     .
     .
     sp_iqsysmon stop_monitor
```

Example 3:

Prints only the RLV and LMA sections of the report:

```
sp_iqsysmon start_monitor,'filemode','-section rlv lma'
     sp_iqsysmon stop_monitor
```

Example 4:

Starts the monitor in file mode and writes statistics for Main Buffer Pool and Memory Manager to the log file every 5 seconds:

```
sp_iqsysmon start_monitor, ‘filemode’, '-interval 5 -section mbufpool memory'
sp_iqsysmon stop_monitor
```

In this section:

[sp_iqsysmon Procedure Examples [page 424]](#)

sp_iqsysmon output examples.

4.3.22.1 sp_iqsysmon Procedure Examples

sp_iqsysmon output examples.

> **i Note**
>
> The output from the statements in the examples are minimized. To expand or recollapse them, click the greater-than symbol (>).

Example 1

This example displays output for the Buffer Allocation (Main and Temporary) after 20 minutes:

```
sp_iqsysmon '00:20:00', 'mbufalloc tbufalloc'
```

```
==============================
Buffer Allocator (Main)"
==============================
STATS-NAME                  VALUE
NActiveCommands                 2
BufAllocMaxBufs              2275( 81.6% )
BufAllocAvailBufs            2115( 93.0% )
BufAllocReserved              160( 7.0% )
BufAllocAvailPF               750( 33.0% )
BufAllocSlots                 100
BufAllocNPinUsers               0
BufAllocNPFUsers                2
BufAllocNPostedUsrs             0
BufAllocNUnpostUsrs             0
BufAllocPinQuota                0
BufAllocNPostEst                0
BufAllocNUnPostEst              0
BufAllocMutexLocks              0
BufAllocMutexWaits              0( 0.0% )
STATS-NAME                  VALUE
NActiveCommands                 2
BufAllocMaxBufs              2275( 81.6% )
BufAllocAvailBufs            2115( 93.0% )
BufAllocReserved              160( 7.0% )
BufAllocAvailPF               750( 33.0% )
BufAllocSlots                 100
BufAllocNPinUsers               0
BufAllocNPFUsers                2
BufAllocNPostedUsrs             0
BufAllocNUnpostUsrs             0
BufAllocPinQuota                0
BufAllocNPostEst                0
BufAllocNUnPostEst              0
BufAllocMutexLocks              0
BufAllocMutexWaits              0( 0.0% )
STATS-NAME                  TOTAL  UNKNWN    HASH        CSORT     ROW
ROWCOL      FP  GARRAY     LOB   BTREE      BM      BV   STORE    TEST
```

```
NumClients                      2       0       0            0       2
0       0       0       0       0       0       0       0       0
PinUserQuota                    0       0       0            0       0
0       0       0       0       0       0       0       0       0
PrefetchUserQuota             160       0       0            0     160
0       0       0       0       0       0       0       0       0
PinUserRegisters                2       2       0            0       0
0       0       0       0       0       0       0       0       0
PfUserRegisters              4697       0       0            0     382
2621     377     182       0       2       0       0       0       0
ClientCountOfPinners            0       1       3            6      10
33      66     100     333     666    1000    3333    6666   10000
Unknown                         0       0       0            0       0
0       0       0       0       0       0       0       0       0
Hash                            0       0       0            0       0
0       0       0       0       0       0       0       0       0
Sort                            0       0       0            0       0
0       0       0       0       0       0       0       0       0
Row                             2       0       0            0       0
0       0       0       0       0       0       0       0       0
RowColumn                       0       0       0            0       0
0       0       0       0       0       0       0       0       0
FP                              0       0       0            0       0
0       0       0       0       0       0       0       0       0
Garray                          0       0       0            0       0
0       0       0       0       0       0       0       0       0
LOB                             0       0       0            0       0
0       0       0       0       0       0       0       0       0
BTree                           0       0       0            0       0
0       0       0       0       0       0       0       0       0
BM                              0       0       0            0       0
0       0       0       0       0       0       0       0       0
BV                              0       0       0            0       0
0       0       0       0       0       0       0       0       0
Store                           0       0       0            0       0
0       0       0       0       0       0       0       0       0
Test                            0       0       0            0       0
0       0       0       0       0       0       0       0       0
DBCC                            0       0       0            0       0
0       0       0       0       0       0       0       0       0
Unknown                         0       0       0            0       0
0       0       0       0       0       0       0       0       0
Unknown                         0       0       0            0       0
0       0       0       0       0       0       0       0       0
Run                             0       0       0            0       0
0       0       0       0       0       0       0       0       0
QCPRun                          0       0       0            0       0
0       0       0       0       0       0       0       0       0
TextDoc                         0       0       0            0       0
0       0       0       0       0       0       0       0       0
Unknown                         0       0       0            0       0
0       0       0       0       0       0       0       0       0
Unknown                         0       0       0            0       0
0       0       0       0       0       0       0       0       0
VDO                             0       0       0            0       0
0       0       0       0       0       0       0       0       0
Load                         Pass       2       0            0       0
0       0       0       0       0       0       0       0       0
STATS-NAME (cont'd)          DBCC  BLKMAP  IQUTIL
NumClients                      0       0       0            0       0
0       0       0       0       0
PinUserQuota                    0       0       0            0       0
0       0       0       0       0
PrefetchUserQuota               0       0       0            0       0
0       0       0       0       0
PinUserRegisters                0       0       0            0       0
0       0       0       0       0
```

```
PfUserRegisters                 0       0       0            0       0
0       0       0    1133       0
ClientCountOfPinners        33333   66666  100000   4294967295
Unknown                         0       0       0            0
Hash                            0       0       0            0
Sort                            0       0       0            0
Row                             0       0       0            0
RowColumn                       0       0       0            0
FP                              0       0       0            0
Garray                          0       0       0            0
LOB                             0       0       0            0
BTree                           0       0       0            0
BM                              0       0       0            0
BV                              0       0       0            0
Store                           0       0       0            0
Test                            0       0       0            0
DBCC                            0       0       0            0
Unknown                         0       0       0            0
Unknown                         0       0       0            0
Run                             0       0       0            0
QCPRun                          0       0       0            0
TextDoc                         0       0       0            0
Unknown                         0       0       0            0
Unknown                         0       0       0            0
VDO                             0       0       0            0
Load                            0       0       0            0       0       0
==============================
Buffer Allocator (Temporary)
==============================
STATS-NAME                  VALUE
NActiveCommands                 2
BufAllocMaxBufs              2275( 81.6% )
BufAllocAvailBufs            2263( 99.5% )
BufAllocReserved               12( 0.5% )
BufAllocAvailPF               908( 39.9% )
BufAllocSlots                 100
BufAllocNPinUsers               2
BufAllocNPFUsers                2
BufAllocNPostedUsrs             0
BufAllocNUnpostUsrs             0
BufAllocPinQuota              175
BufAllocNPostEst                2
BufAllocNUnPostEst              2
BufAllocMutexLocks              0
BufAllocMutexWaits              0( 0.0% )
STATS-NAME                  TOTAL  UNKNWN    HASH        CSORT     ROW
ROWCOL      FP  GARRAY     LOB   BTREE      BM      BV   STORE    TEST
NumClients                      4       0       0            4       0
0       0       0       0       0       0       0       0       0
PinUserQuota                   10       0       0           10       0
0       0       0       0       0       0       0       0       0
PrefetchUserQuota               2       0       0            2       0
0       0       0       0       0       0       0       0       0
PinUserRegisters              668       0     300          247       0
0       0       0       0       0       0       0       0       0
PfUserRegisters               675       0       0          295       0
0       0       0       0       0       0       0       1       0
ClientCountOfPinners            0       1       3            6      10
33      66     100     333     666    1000    3333    6666   10000
Unknown                         0       0       0            0       0
0       0       0       0       0       0       0       0       0
Hash                            0       0       0            0       0
0       0       0       0       0       0       0       0       0
Sort                            2       0       1            0       1
0       0       0       0       0       0       0       0       0
Row                             0       0       0            0       0
0       0       0       0       0       0       0       0       0
```

```
RowColumn                       0       0       0            0       0
0       0       0       0       0       0       0       0       0
FP                              0       0       0            0       0
0       0       0       0       0       0       0       0       0
Garray                          0       0       0            0       0
0       0       0       0       0       0       0       0       0
LOB                             0       0       0            0       0
0       0       0       0       0       0       0       0       0
BTree                           0       0       0            0       0
0       0       0       0       0       0       0       0       0
BM                              0       0       0            0       0
0       0       0       0       0       0       0       0       0
BV                              0       0       0            0       0
0       0       0       0       0       0       0       0       0
Store                           0       0       0            0       0
0       0       0       0       0       0       0       0       0
Test                            0       0       0            0       0
0       0       0       0       0       0       0       0       0
DBCC                            0       0       0            0       0
0       0       0       0       0       0       0       0       0
Unknown                         0       0       0            0       0
0       0       0       0       0       0       0       0       0
Unknown                         0       0       0            0       0
0       0       0       0       0       0       0       0       0
Run                             0       0       0            0       0
0       0       0       0       0       0       0       0       0
QCPRun                         `        0       0            0       0
0       0       0       0       0       0       0       0       0
TextDoc                         0       0       0            0       0
0       0       0       0       0       0       0       0       0
Unknown                         0       0       0            0       0
0       0       0       0       0       0       0       0       0
Unknown                         0       0       0            0       0
0       0       0       0       0       0       0       0       0
VDO                             0       0       0            0       0
0       0       0       0       0       0       0       0       0
Load                         Pass       2       0            0       0
0       0       0       0       0       0       0       0       0
STATS-NAME (cont'd)          DBCC  BLKMAP  IQUTIL
NumClients                      0       0       0            0       0
0       0       0       0       0
PinUserQuota                    0       0       0            0       0
0       0       0       0       0
PrefetchUserQuota               0       0       0            0       0
0       0       0       0       0
PinUserRegisters                0       0       0          110       2
0       0       0       0       9
PfUserRegisters                 0       0       0          378       0
0       0       1       0       0
ClientCountOfPinners        33333   66666  100000   4294967295
Unknown                         0       0       0            0
Hash                            0       0       0            0
Sort                            0       0       0            0
Row                             0       0       0            0
RowColumn                       0       0       0            0
FP                              0       0       0            0
Garray                          0       0       0            0
LOB                             0       0       0            0
BTree                           0       0       0            0
BM                              0       0       0            0
BV                              0       0       0            0
Store                           0       0       0            0
Test                            0       0       0            0
DBCC                            0       0       0            0
Unknown                         0       0       0            0
Unknown                         0       0       0            0
Run                             0       0       0            0
QCPRun                          0       0       0            0
```

```
TextDoc                         0       0       0            0
Unknown                         0       0       0            0
Unknown                         0       0       0            0
VDO                             0       0       0            0
Load                            0       0       0            0       0       0
```

Example 2

This example displays output for the Buffer Manager (Main and Temporary) after 20 minutes:

```
sp_iqsysmon '00:20:00', 'mbufman tbufman'
```

```
==============================
Buffer Manager (Main)
==============================
STATS-NAME                  TOTAL    NONE  TXTPOS  TXTDOC  CMPACT  BTREEV
BTREEF      BV     VDO   DBEXT    DBID    SORT   STORE  GARRAY
Finds                       80137       0       0       0       0    9046
3307       0   20829       0       0       0       0     275
Hits                        80090       0       0       0       0    9015
3291       0   20829       0       0       0       0     275
Hit%                         99.9       0       0       0       0    99.7
99.5       0     100       0       0       0       0     100
FalseMiss                   26469       0       0       0       0      63
40       0    1097       0       0       0       0       0
UnOwnRR                        48       0       0       0       0      31
16       0       1       0       0       0       0       0
Cloned                          0       0       0       0       0       0
0       0       0       0       0       0       0       0
Creates                      1557       0       0       0       0      60
179       0     256       0       0       0       0      58
Destroys                      546       0       0       0       0      12
21       0       6       0       0       0       0      29
Dirties                      7554       0       0       0       0    1578
585       0       0       0       0       0       0       0
RealDirties                  2254       0       0       0       0     117
180       0     542       0       0       0       0      58
PrefetchReqs                   80       0       0       0       0       0
0       0      74       0       0       0       0       0
PrefetchNotInMem                1       0       0       0       0       0
0       0       1       0       0       0       0       0
PrefetchInMem                1466       0       0       0       0       0
0       0    1466       0       0       0       0       0
Reads                          48       0       0       0       0      31
16       0       1       0       0       0       0       0
PReadBlks                     114       0       0       0       0      80
32       0       2       0       0       0       0       0
PReadKB                         0       0       0       0       0       0
0       0       0       0       0       0       0       0
ReReads                         0       0       0       0       0       0
0       0       0       0       0       0       0       0
Writes                       2002       0       0       0       0     104
163       0     538       0       0       0       0      29
PWriteBlks                   6506       0       0       0       0     210
326       0    1115       0       0       0       0      58
PWriteKB                        0       0       0       0       0       0
0       0       0       0       0       0       0       0
GrabbedDirty                    0       0       0       0       0       0
0       0       0       0       0       0       0       0
ReadRemoteRpc                   0       0       0       0       0       0
0       0       0       0       0       0       0       0
ReadRemotePhyIO                 0       0       0       0       0       0
0       0       0       0       0       0       0       0
```

```
STATS-NAME (cont'd)        BARRAY  BLKMAP    HASH    CKPT      BM    TEST
CMID   RIDCA     LOB  LVCRID    FILE  RIDMAP   RVLOG
Finds                        2681    8329       0       0   35670       0
0       0       0       0       0       0       0
Hits                         2681    8329       0       0   35670       0
0       0       0       0       0       0       0
Hit%                          100     100       0       0     100       0
0       0       0       0       0       0       0
FalseMiss                      84    8329       0       0   16856       0
0       0       0       0       0       0       0
UnOwnRR                         0       0       0       0       0       0
0       0       0       0       0       0       0
Cloned                          0       0       0       0       0       0
0       0       0       0       0       0       0
Creates                       108     358       0       0     538       0
0       0       0       0       0       0       0
Destroys                        0     126       0       0      59       0
0       0       0       0       0       0       0
Dirties                       512     235       0       0    4644       0
0       0       0       0       0       0       0
RealDirties                   128     593       0       0     636       0
0       0       0       0       0       0       0
PrefetchReqs                    6       0       0       0       0       0
0       0       0       0       0       0       0
PrefetchNotInMem                0       0       0       0       0       0
0       0       0       0       0       0       0
PrefetchInMem                   0       0       0       0       0       0
0       0       0       0       0       0       0
Reads                           0       0       0       0       0       0
0       0       0       0       0       0       0
PReadBlks                       0       0       0       0       0       0
0       0       0       0       0       0       0
PReadKB                         0       0       0       0       0       0
0       0       0       0       0       0       0
ReReads                         0       0       0       0       0       0
0       0       0       0       0       0       0
Writes                        128     466       0       0     574       0
0       0       0       0       0       0       0
PWriteBlks                    239    3728       0       0     830       0
0       0       0       0       0       0       0
PWriteKB                        0       0       0       0       0       0
0       0       0       0       0       0       0
GrabbedDirty                    0       0       0       0       0       0
0       0       0       0       0       0       0
ReadRemoteRpc                   0       0       0       0       0       0
0       0       0       0       0       0       0
ReadRemotePhyIO                 0       0       0       0       0       0
0       0       0       0       0       0       0
STATS-NAME                  VALUE
BusyWaits                      98
LRUNumLocks                401784
LRUNumSpinsWoTO                 0      0%
LRUNumSpinLoops              4315
LRUNumTimeOuts               4315  -1.10%
BmapHTNumLocks                  0
BmapHTNumWaits                  0      0%
CacheTeamTimesWoken           182
CacheTeamNumAsleep             10
BmapHTMaxEntries             4096
BmapHTNEntries                 27
BmapHTNInserts              31954
BmapHTNCollisn                203
BmapHTNFinds                51419
BmapHTNHits                 19576
BmapHTNHits1                19550
BmapHTNHits2                   26
BmapHTNClears               31933
BmapHTNLChain                   1
```

```
BmapHTNRehash                   0
BlockmapMutexsNLocks            0
BlockmapMutexsNWaits            0
BlockmapUID                  3659
BlockmapUIDnallocs           3652
BlockmapRegEver             31851
BlockmapRegisters           31844
BufHTNBuckets                4608
BufHTNEntries                1208
BufHTNw2orMore                158
BufHTMaxBucketSize             19
BufHTNFoiledOps                 0
IONumLocks                      0
IONumWaits                      0      0%
==============================
Buffer Manager (Temporary)
==============================
STATS-NAME                  TOTAL    NONE  TXTPOS  TXTDOC  CMPACT  BTREEV
BTREEF      BV     VDO   DBEXT    DBID    SORT   STORE  GARRAY
Finds                       31656       0       0       0       0       0
0       0       0       0       0    1022       0       0
Hits                        31655       0       0       0       0       0
0       0       0       0       0    1022       0       0
Hit%                          100       0       0       0       0       0
0       0       0       0       0     100       0       0
FalseMiss                   23898       0       0       0       0       0
0       0       0       0       0       0       0       0
UnOwnRR                         0       0       0       0       0       0
0       0       0       0       0       0       0       0
Cloned                          0       0       0       0       0       0
0       0       0       0       0       0       0       0
Creates                      5682       0       0       0       0       0
0       0       0       0       0    1048     716       0
Destroys                     5670       0       0       0       0       0
0       0       0       0       0     821      17       0
Dirties                      6702       0       0       0       0       0
0       0       0       0       0     379       0       0
RealDirties                  5692       0       0       0       0       0
0       0       0       0       0    1048     716       0
PrefetchReqs                    1       0       0       0       0       0
0       0       0       0       0       0       0       0
PrefetchNotInMem                1       0       0       0       0       0
0       0       0       0       0       0       0       0
PrefetchInMem                 446       0       0       0       0       0
0       0       0       0       0     446       0       0
Reads                           2       0       0       0       0       0
0       0       0       0       0       0       0       0
PReadBlks                    4096       0       0       0       0       0
0       0       0       0       0       0       0       0
PReadKB                         0       0       0       0       0       0
0       0       0       0       0       0       0       0
ReReads                         2       0       0       0       0       0
0       0       0       0       0       0       0       0
Writes                         10       0       0       0       0       0
0       0       0       0       0       0       0       0
PWriteBlks                     80       0       0       0       0       0
0       0       0       0       0       0       0       0
PWriteKB                        0       0       0       0       0       0
0       0       0       0       0       0       0       0
GrabbedDirty                    0       0       0       0       0       0
0       0       0       0       0       0       0       0
ReadRemoteRpc                   0       0       0       0       0       0
0       0       0       0       0       0       0       0
ReadRemotePhyIO                 0       0       0       0       0       0
0       0       0       0       0       0       0       0
STATS-NAME (cont'd)        BARRAY  BLKMAP    HASH    CKPT      BM    TEST
CMID   RIDCA     LOB  LVCRID    FILE  RIDMAP   RVLOG
```

```
Finds                           0    8569     124       0   21939       0
0       0       0       0       2       0       0
Hits                            0    8569     124       0   21939       0
0       0       0       0       1       0       0
Hit%                            0     100     100       0     100       0
0       0       0       0      50       0       0
FalseMiss                       0    8569       0       0   15328       0
0       0       0       0       1       0       0
UnOwnRR                         0       0       0       0       0       0
0       0       0       0       0       0       0
Cloned                          0       0       0       0       0       0
0       0       0       0       0       0       0
Creates                         0    1440     777       0    1041       0
0       0       0       0       0     660       0
Destroys                        0    1434     777       0     123       0
0       0       0       0       0     660       0
Dirties                         0       0       0       0    6323       0
0       0       0       0       0       0       0
RealDirties                     0    1440     777       0    1051       0
0       0       0       0       0     660       0
PrefetchReqs                    0       0       0       0       0       0
0       0       0       0       1       0       0
PrefetchNotInMem                0       0       0       0       0       0
0       0       0       0       1       0       0
PrefetchInMem                   0       0       0       0       0       0
0       0       0       0       0       0       0
Reads                           0       0       0       0       0       0
0       0       0       0       2       0       0
PReadBlks                       0       0       0       0       0       0
0       0       0       0    4096       0       0
PReadKB                         0       0       0       0       0       0
0       0       0       0       0       0       0
ReReads                         0       0       0       0       0       0
0       0       0       0       2       0       0
Writes                          0       0       0       0      10       0
0       0       0       0       0       0       0
PWriteBlks                      0       0       0       0      80       0
0       0       0       0       0       0       0
PWriteKB                        0       0       0       0       0       0
0       0       0       0       0       0       0
GrabbedDirty                    0       0       0       0       0       0
0       0       0       0       0       0       0
ReadRemoteRpc                   0       0       0       0       0       0
0       0       0       0       0       0       0
ReadRemotePhyIO                 0       0       0       0       0       0
0       0       0       0       0       0       0
STATS-NAME                  VALUE
BusyWaits                       0
LRUNumLocks                136253
LRUNumSpinsWoTO                 0      0%
LRUNumSpinLoops              2780
LRUNumTimeOuts               2780  -0.02%
BmapHTNumLocks                  0
BmapHTNumWaits                  0      0%
CacheTeamTimesWoken             1
CacheTeamNumAsleep             10
BmapHTMaxEntries             4096
BmapHTNEntries                 17
BmapHTNInserts               2334
BmapHTNCollisn                  0
BmapHTNFinds                  183
BmapHTNHits                     0
BmapHTNHits1                    0
BmapHTNHits2                    0
BmapHTNClears                2327
BmapHTNLChain                   0
BmapHTNRehash                   0
BlockmapMutexsNLocks            0
```

```
BlockmapMutexsNWaits            0
BlockmapUID                  2380
BlockmapUIDnallocs           2335
BlockmapRegEver              2344
BlockmapRegisters            2334
BufHTNBuckets                4608
BufHTNEntries                  24
BufHTNw2orMore                  0
BufHTMaxBucketSize              3
BufHTNFoiledOps                 0
IONumLocks                      0
IONumWaits                      0      0%
```

Example 3

This example displays output for the Buffer Pool (Main and Temporary) after 20 minutes:

```
sp_iqsysmon '00:20:00', 'mbufpool tbufpool'
```

```
==============================
Buffer Pool (Main)
==============================

STATS-NAME                  TOTAL    NONE  TXTPOS  TXTDOC  CMPACT  BTREEV
BTREEF      BV     VDO   DBEXT    DBID    SORT   STORE  GARRAY
MovedToMRU                  68731       0       0       0       0    9094
2767       0   21083       0       0       0       0     303
MovedToWash                     0       0       0       0       0       0
0       0       0       0       0       0       0       0
RemovedFromLRU              67564       0       0       0       0    9020
2597       0   20830       0       0       0       0     274
RemovedFromWash             11457       0       0       0       0    1559
356       0    2189       0       0       0       0      68
RemovedInScanMode               0       0       0       0       0       0
0       0       0       0       0       0       0       0
MovedToPSList                   0       0       0       0       0       0
0       0       0       0       0       0       0       0
RemovedFromPSList               0       0       0       0       0       0
0       0       0       0       0       0       0       0
STATS-NAME (cont'd)        BARRAY  BLKMAP    HASH    CKPT      BM    TEST
CMID   RIDCA     LOB  LVCRID    FILE  RIDMAP   RVLOG
MovedToMRU                   2169    8561       0       0   24754       0
0       0       0       0       0       0       0
MovedToWash                     0       0       0       0       0       0
0       0       0       0       0       0       0
RemovedFromLRU               2065    8330       0       0   24448       0
0       0       0       0       0       0       0
RemovedFromWash               233    1437       0       0    5615       0
0       0       0       0       0       0       0
RemovedInScanMode               0       0       0       0       0       0
0       0       0       0       0       0       0
MovedToPSList                   0       0       0       0       0       0
0       0       0       0       0       0       0
RemovedFromPSList               0       0       0       0       0       0
0       0       0       0       0       0       0
STATS-NAME                  VALUE
Pages                        2787
InUse                        1208 ( 43.3% )
Dirty                          11 ( 0.4% )
Pinned                         19 ( 0.7% )
Flushes                         0
FlushedBufferCount              0
```

```
GetPageFrame                 1605
GetPageFrameFailure             0
GotEmptyFrame                1605
Washed                          0
TimesSweepersWoken              0
PriorityWashed                  0
NPrioritySweepersWoken          0
washTeamSize                   10
WashMaxSize                   455 ( 16.3% )
washNBuffers                  455 ( 16.3% )
washNDirtyBuffers               0 ( 0.0% )
washSignalThreshold            46 ( 1.7% )
washNActiveSweepers             0
NPriorityWashBuffers            0
NActivePrioritySweepers         0
washIntensity                   0
FlushAndEmpties                 0
EmptiedBufferCount              0
EmptiedSkippedCount             0
EmptiedWriteCount               0
EmptiedErrorCount               0
nAffinityTotal                  0 ( 0.0% )
nAffinityArea                   0 ( 0.0% )
==============================
Buffer Pool (Temporary)
==============================

STATS-NAME                  TOTAL    NONE  TXTPOS  TXTDOC  CMPACT  BTREEV
BTREEF      BV     VDO   DBEXT    DBID    SORT   STORE  GARRAY
MovedToMRU                  30514       0       0       0       0       0
0       0       0       0       0    1218     696       0
MovedToWash                   258       0       0       0       0       0
0       0       0       0       0       0     256       0
RemovedFromLRU              30506       0       0       0       0       0
0       0       0       0       0    1218     694       0
RemovedFromWash             30503       0       0       0       0       0
0       0       0       0       0    1218     694       0
RemovedInScanMode               0       0       0       0       0       0
0       0       0       0       0       0       0       0
MovedToPSList                   0       0       0       0       0       0
0       0       0       0       0       0       0       0
RemovedFromPSList               0       0       0       0       0       0
0       0       0       0       0       0       0       0
STATS-NAME (cont'd)        BARRAY  BLKMAP    HASH    CKPT      BM    TEST
CMID   RIDCA     LOB  LVCRID    FILE  RIDMAP   RVLOG
MovedToMRU                      0    8575     124       0   19898       0
0       0       0       0       3       0       0
MovedToWash                     0       0       0       0       0       0
0       0       0       0       2       0       0
RemovedFromLRU                  0    8569     124       0   19898       0
0       0       0       0       3       0       0
RemovedFromWash                 0    8569     124       0   19898       0
0       0       0       0       0       0       0
RemovedInScanMode               0       0       0       0       0       0
0       0       0       0       0       0       0
MovedToPSList                   0       0       0       0       0       0
0       0       0       0       0       0       0
RemovedFromPSList               0       0       0       0       0       0
0       0       0       0       0       0       0
STATS-NAME                 VALUE
Pages                        2787
InUse                          24 ( 0.9% )
Dirty                          17 ( 0.6% )
Pinned                          4 ( 0.1% )
Flushes                         0
FlushedBufferCount              0
GetPageFrame                 5684
GetPageFrameFailure             0
```

```
GotEmptyFrame                5684
Washed                          0
TimesSweepersWoken              0
PriorityWashed                  0
NPrioritySweepersWoken          0
washTeamSize                   10
WashMaxSize                   455 ( 16.3% )
washNBuffers                   20 ( 0.7% )
washNDirtyBuffers              13 ( 0.5% )
washSignalThreshold            46 ( 1.7% )
washNActiveSweepers             0
NPriorityWashBuffers            0
NActivePrioritySweepers         0
washIntensity                   0
FlushAndEmpties                 0
EmptiedBufferCount              0
EmptiedSkippedCount             0
EmptiedWriteCount               0
EmptiedErrorCount               0
nAffinityTotal                  0 ( 0.0% )
nAffinityArea                   0 ( 0.0% )
```

Example 4

This example displays output for the Prefetch Manager (Main and Temporary) after 20 minutes:

```
sp_iqsysmon '00:20:00', 'mprefetch tprefetch'
```

```
==============================
Prefetch Manager (Main)
==============================
STATS-NAME                  VALUE
PFMgrNThreads                  10
PFMgrNSubmitted                81
PFMgrNDropped                   0
PFMgrNValid                     0
PFMgrNRead                      1
PFMgrNReading                   0
PFMgrCondVar                Locks  0  Lock-Waits 0 ( 0.0% )  Signals 0
Broadcasts  2  Waits  2
==============================
Prefetch Manager (Temporary)
==============================
STATS-NAME                  VALUE
PFMgrNThreads                  10
PFMgrNSubmitted                 1
PFMgrNDropped                   0
PFMgrNValid                     0
PFMgrNRead                      1
PFMgrNReading                   0
PFMgrCondVar                Locks  0  Lock-Waits 0 ( 0.0% )  Signals 0
Broadcasts  2  Waits  2
```

Example 5

This example displays output for the IQ Store Free List (Main and Temporary) after 20 minutes:

```
sp_iqsysmon '00:20:00', 'mfreelist tfreelist'
```

```
==============================
IQ Store (Main) Free List
==============================
STATS-NAME                  VALUE
FLBitCount                  74036
FLIsOutOfSpace                NO
FLMutexLocks                    0
FLMutexWaits                    0 ( 0.0% )
==============================
IQ Store (Temporary) Free List
==============================
STATS-NAME                  VALUE
FLBitCount                   4784
FLIsOutOfSpace                NO
FLMutexLocks                    0
FLMutexWaits                    0 ( 0.0% )
```

Example 6

This example displays output for Memory Manager, Thread Manger, CPU utilization, Transaction Manager after 20 minutes:

```
sp_iqsysmon '00:20:00', 'memory threads cpu txn'
```

```
==============================
Memory Manager
==============================
STATS-NAME                      VALUE
MemAllocated                 67599536  ( 66015 KB )
MemAllocatedMax             160044816 ( 156293 KB )
MemAllocatedEver           1009672456 ( 986008 KB )
MemNAllocated                   77309
MemNAllocatedEver              914028
MemNTimesLocked                     0
MemNTimesWaited                     0      ( 0.0 %)
==============================
Thread Manager
==============================

STATS-NAME                      VALUE
ThrNumOfCpus                        4
ThreadLimit                        99
ThrNumThreads                      98     ( 99.0 %)
ThrReserved                        15     ( 15.2 %)
ThrNumFree                         55     ( 55.6 %)
NumThrUsed                         44     ( 44.4 %)
UsedPerActiveCmd                   22
ThrNTeamsInUse                      5
ThrMaxTeams                         7
NumTeamsAlloc                     238
TeamThrAlloc                      421
SingleThrAlloc                    492
ThrMutexLocks                       0
```

```
ThrMutexWaits                       0      ( 0.0 %)
==============================
CPU time statistics
==============================
STATS-NAME                      VALUE
Elapsed Seconds                 59.65     ( 25.0 %)
CPU User Seconds                37.79     ( 15.8 %)
CPU Sys Seconds                  1.89      ( 0.8 %)
CPU Total Seconds               39.68     ( 16.6 %)
==============================
Transaction Manager
==============================
STATS-NAME                      VALUE
TxnMgrNPending                      0
TxnMgrNBlocked                      2
TxnMgrNWaiting                      0
TxnMgrPCcondvar                 Locks     0       Lock-Wait 0 ( 0.0 %)  Signals
0  Broadcasts 2 Waits 2
TxnMgrTxnIDseq                    407
TxnMgrtxncblock                 Locks     0       Lock-Wait 0 ( 0.0 %)
TxnMgrVersionID                     0
TxnMgrOAVI                          0
TxnMgrVersionLock               Locks     0       Lock-Wait 0 ( 0.0 %)  Signals
0  Broadcasts 0 Waits 0
```

Example 7

This example displays output for server context and catalog statistics after 20 minutes:

```
sp_iqsysmon '00:20:00', 'context catalog'
```

```
==============================
Context Server statistics
==============================
STATS-NAME                  VALUE
StCntxNumConns                  1
StCntxNResource                16
StCntxNOrigResource            18
StCntxNWaiting                  0
StCntxNWaited                   0
StCntxNAdmitted              1116
StCntxLock                  Locks  0 Lock-Waits  0 ( 0.0 %)
StCntxCondVar               Locks  0 Lock-Waits  0 ( 0.0 %)
==============================
Catalog, DB Log, and Repository statistics
==============================
STATS-NAME                  VALUE
CatalogLock               RdLocks  0    RdWaits  0 ( 0.0 %)  RdTryFails 0
WrLocks  30037  WrWaits  0 ( 0.0 %)  WrTryFail 0
DbLogMLock                  Locks  0 Lock-Waits  0 ( 0.0 %)
DbLogSLock                  Locks  0 Lock-Waits  0 ( 0.0 %)
RepositoryNList                 0
RepositoryLock              Locks  1  SpinsWoTO  0 ( 0.0 %)      Spins  0
TimeOuts     0 ( 0.0 %)
```

Example 8

This example displays output for IQ RLV In-Memory Store and Large Memory Allocator (LMA) statistics after 20 minutes:

```
sp_iqsysmon '00:20:00', 'rlv lma'
```

```
==============================
IQ In-Memory Store
==============================
STATS-NAME                VALUE
RLV Memory Limit        2048 MB
RLV Memory Used            0 MB
RLV Chunks Used               0
==============================
Large Memory Allocator
==============================
STATS-NAME                VALUE
Large Memory Space      2048 MB
Large Memory Max Fle     512 MB
Large Memory Num Fle          0
Large Memory Flexibl        0.5
Large Memory Flexibl       0 MB
Large Memory Inflexi        0.9
Large Memory Inflexi       0 MB
Large Memory Anti-St        0.5
Large Memory Num Con          0
```

4.3.23 sp_iqpassword Procedure

Changes a user’s password.

> **Note**
>
> Though `sp_iqpassword` is still supported for backwards compatibility, use `ALTER USER` to change a user password.

Syntax 1

```
call sp_iqpassword (‘<caller_password>’, ‘<new_password>’ [, ‘<user_name>’])
```

Syntax 2

```
sp_iqpassword ‘<caller_password>’, ‘<new_password>’ [, ‘<user_ name>’]
```

Parameters

caller_password Your password. When you are changing your own password, this is your old password. When a user with the CHANGE PASSWORD system privilege is changing another user's password, caller_password is the password of the user making the change.

new_password New password for the user, or for `<loginname>`.

user_name Login name of the user whose password is being changed by by another user with CHANGE PASSWORD system privilege. Do not specify user_name when changing your own password.

Privileges

You must have EXECUTE privilege on the system procedure. No additional system privilege is need to set your own password. You need the CHANGE PASSWORD system privilege to set other users' passwords.

Remarks

A user password is an identifier. Any user can change his or her own password using `sp_iqpassword`. The CHANGE PASSWORD system privilege is required to change the password of any existing user.

Identifiers have a maximum length of 128 bytes. They must be enclosed in double quotes or square brackets if any of these conditions are true:

- The identifier contains spaces.
- The first character of the identifier is not an alphabetic character (as defined below).
- The identifier contains a reserved word.
- The identifier contains characters other than alphabetic characters and digits.
  Alphabetic characters include the alphabet, as well as the underscore character (_), at sign (@), number sign (#), and dollar sign ($). The database collation sequence dictates which characters are considered alphabetic or digit characters.

Example

Changes the password of the logged-in user from irk103 to exP984:

```
sp_iqpassword 'irk103', 'exP984'
```

If the logged-in user has the CHANGE PASSWORD system privilege or `joe`, the password of user `joe` from `eprr45` to `pdi032`:

```
call sp_iqpassword ('eprr45', 'pdi932', 'joe')
```

4.3.24 sp_objectpermission System Procedure

Generates a report on object privileges granted to the specified role, or user name, or the object privileges granted on the specified object or dbspace.

Syntax

```
sp_objectpermission ( [<object_name>], [<object_owner>], [<object_type>] )
```

Parameters

object_name The name of an object or dbspace or a user or a role. If not specified, object privileges of the current user are reported. Default value is NULL.

object_owner The name of the object owner for the specified object name. The object privileges of the specified object owned by the specified object owner are displayed. This parameter must be specified to obtain the object privileges of an object owned by another user or role. Default value is NULL.

object_type Valid values are:

- TABLE – column-level object privileges also appear.
- VIEW
- MATERIALIZED VIEW
- SEQUENCE
- PROCEDURE
- FUNCTION
- DBSPACE
- USER

If no value is specified, privileges on all object types are returned. Default value is NULL.

Privileges

You must have EXECUTE privilege on the system procedure.. Any user can execute `sp_objectpermission` to obtain all the object privileges granted to him- or herself. Object owners can also execute this procedure to obtain the object privileges for self-owned objects. Additional system privileges are needed to obtain object privileges for the following:

Object privileges granted to other users or granted on objects owned by other users You must also have the MANAGE ANY OBJECT PRIVILEGE system privilege

Object privileges that are granted on objects owned by a role or granted to a role You must also have the MANAGE ANY OBJECT PRIVILEGE system privilege or be a role administrator on the role

Object privileges of a dbspace You must have the MANAGE ANY DBSPACE system privilege

Remarks

Column Name	Data Type	Description
grantor	char(128)	The user ID of the grantor
grantee	char(128)	The user ID of the grantee
object_name	char(128)	The name of the object
owner	char(128)	The name of the object owner
object_type	char(20)	The type of object
column_name	char(128)	The name of the column
permission	char(20)	The name of the privilege
grantable	char(1)	Whether or not the privilege is grantable

All arguments are optional and can generate these reports:

- If input is an object (table, view, procedure, function, sequence, and so on), procedure displays list of all roles and user that have different object privilege on the object.
- If input is a role or user, procedure displays list of all object privileges granted to the role or input. When executing `sp_objectpermission` to display object privileges of a user or a role, the object privileges that are inherited through role grants also.
- If input is a dbspace name, procedure displays list of all user or roles that have CREATE privilege on the specified dbspace.
- By default, object type is NULL and the object privileges for all existing object types matching the specified object name appear.

Example

The following GRANT statements are executed:

```
GRANT SERVER OPERATOR TO r4;
GRANT BACKUP DATABASE TO r3 WITH ADMIN OPTION;
GRANT DROP CONNECTION TO r3 WITH ADMIN ONLY OPTION;
GRANT MONITOR TO r2;GRANT CHECKPOINT TO r1;
GRANT ROLE r2 TO r1 WITH ADMIN OPTION;
GRANT ROLE r3 TO r2 WITH NO ADMIN OPTION;
GRANT ROLE r4 TO r3 WITH ADMIN ONLY OPTION;
```

Consider these object privileges:

- `r5` owns a table named `test_tab` and a procedure named `test_proc` in the database.
- `u5`, which has administrative rights over `r5`, grants the following privileges:
    - GRANT SELECT ON r5.test_tab TO r2 WITH GRANT OPTION;
    - GRANT SELECT (c1), UPDATE (c1) ON r5.test_tab TO r6 WITH GRANT OPTION;
    - GRANT EXECUTE ON r5.test_proc TO r3;
- `u6`, which has administrative rights over `r6`, grants the following privileges:
    - GRANT SELECT (c1), REFERENCES (c1) ON r5.test_tab TO r3;

If `sp_objectpermission( 'r1' )` is executed, output is similar to:

Table 18: Example sp_objectpermission('r1') Output

grantor	grantee	object_name
u5	r2	test_tab
u6	r3	test_tab
u6	r3	test_tab
u6	r3	test_proc

(Continued)

owner	object_type	grantor
r5	TABLE	u5
r5	COLUMN	u6
r5	COLUMN	u6
r5	PROCEDURE	u6

(Continued)

grantable	column_name	privilege
Y	NULL	SELECT
N	c1	SELECT
Y	c1	REFERENCES
N	NULL	EXECUTE

If `sp_objectpermission( 'test_tab', 'r5', 'table' )` is executed, output is similar to:

Table 19: Example sp_objectpermission('test_tab', 'r5', 'table') Output

grantor	grantee	object_name
u5	r2	test_tab
u5	r6	test_tab
u5	r6	test_tab
u6	r3	test_tab
u6	r3	test_tab

(Continued)

owner	object_type	grantor
r5	TABLE	u5
r5	COLUMN	u5
r5	COLUMN	u5
r5	COLUMN	u6
r5	COLUMN	u6

(Continued)

column_name	privilege	grantable
NULL	SELECT	Y
c1	SELECT	Y
c1	UPDATE	Y
c1	SELECT	N
c1	REFERENCES	N

4.3.25 sp_sys_priv_role_info System Procedure

Generates a report to map a system privilege to the corresponding system role. A single row is returned for each system privilege.

Syntax

```
sp_sys_priv_role_info()
```

Privileges

You must have EXECUTE privilege on the system procedure.

Remarks

Column Name	Data Type	Description
sys_priv_name	char(128)	The name of the system privilege
sys_priv_role_name	char(128)	The role name corresponding to the system privilege.
sys_priv_id	unsigned int	The id of the system privilege.

4.3.26 sp_alter_secure_feature_key System Procedure

Alters a previously-defined secure feature key by modifying the authorization key and/or the feature list.

Syntax

```
sp_alter_secure_feature_key (
    <name>,
    <auth_key>,
    <features> )
```

Parameters

name the VARCHAR (128) name for the secure feature key you want to alter. A key with the given name must already exist.

auth_key the CHAR (128) authorization key for the secure feature key. The authorization key must be either a non-empty string of at least six characters, or NULL, indicating that the existing authorization key is not to be changed.

features the LONG VARCHAR, comma-separated list of secure features that the key can enable. The feature_list can be NULL, indicating that the existing feature_list is not to be changed.

Privileges

You must have EXECUTE privilege on the system procedure. In addition, you must be the database server owner and have the manage_keys feature enabled on the connection.

Remarks

This procedure allows you to alter the authorization key or feature list of an existing secure feature key.

4.3.27 sp_create_secure_feature_key System Procedure

Creates a new secure feature key.

Syntax

```
sp_create_secure_feature_key (
    <name>,
    <auth_key>,
    <features> )
```

Parameters

name the VARCHAR (128) name for the new secure feature key. This argument cannot be NULL or an empty string.

auth_key the CHAR (128) authorization key for the secure feature key. The authorization key must be a non-empty string of at least six characters.

features the LONG VARCHAR comma-separated list of secure features that the new key can enable. Specifying "-" before a feature means that the feature is not re-enabled when the secure feature key is set.

Privileges

You must have EXECUTE privilege on the system procedure. In addition, you must be the database server owner and have the manage_keys feature enabled on the connection.

Remarks

This procedure creates a new secure feature key that can be given to any user. The system secure feature key is created using the -sk database server option.

4.3.28 sp_drop_secure_feature_key System Procedure

Deletes a secure feature key.

Syntax

```
sp_drop_secure_feature_key ( <name> )
```

Parameters

name the VARCHAR (128) name of the secure feature key to drop.

Privileges

You must have EXECUTE privilege on the system procedure. In addition, you must be the database server owner and have the manage_keys feature enabled on the connection.

Remarks

If the named key does not exist, an error is returned. If the named key exists, it is deleted as long as it is not the last secure feature key that is allowed to manage secure features and secure feature keys. For example, the system secure feature key cannot be dropped until there is another key that has the manage_features and manage_keys secure features enabled.

4.3.29 sp_list_secure_feature_keys System Procedure

Returns information about the contents of a directory.

Syntax

```
sp_list_secure_feature_keys ( )
```

Privileges

You must have EXECUTE privilege on the system procedure. In addition, you must be the database server owner and have the manage_keys feature enabled on the connection.

Remarks

Column Name	Data Type	Description
name	VARCHAR(128)	The name of the secure feature key.
features	LONG VARCHAR	The secure features enabled by the secure feature key.

This procedure returns the names of existing secure feature keys, as well as the set of secure features that can be enabled by each key.

If the user has the manage_features and manage_keys secure features enabled, then the procedure returns a list of all secure feature keys.

If the user only has the manage_keys secure feature enabled, then the procedure returns keys that have the same features or a subset of the same features that the current user has enabled.

4.3.30 sp_use_secure_feature_key System Procedure

Enables an existing secure feature key.

Syntax

```
sp_use_secure_feature_key ( <name>, <sfkey>)
```

Parameteer

name the VARCHAR (128) name of the secure feature key to be enabled.

sfkey the CHAR (128) authorization key for the secure feature key being enabled. The authorization key must be at least six characters.

Privileges

You must have EXECUTE privilege on the system procedure.

Remarks

This procedure enables the secure features that are turned on by the specified secure feature key.

5 Appendix: Startup and Connection Parameters

Reference material for startup options and connection parameters for the `start_iq` utility.

In this section:

5.1 -ec database server option

Uses transport layer security or simple encryption to encrypt all command sequence communication protocol packets (such as DBLib and ODBC) transmitted to and from all clients.

Syntax

```
start_iq -ec <encryption-options> ...
```

```
<encryption-options> :
{ NONE |
  SIMPLE |
  TLS ( [ FIPS={ Y | N }; ]
          IDENTITY=<server-identity-filename>;
          IDENTITY_PASSWORD=<password> ) }, ...
```

Allowed values

NONE

The database server accepts connections that aren't encrypted.

SIMPLE

The database server accepts connections that are encrypted using simple encryption. This type of encryption is supported on all platforms, and on previous versions of the database server and clients. Simple encryption doesn't provide server authentication, RSA encryption, or other features of transport layer security.

TLS

The database server accepts connections that are encrypted using the RSA encryption algorithm. The TLS parameter accepts the following arguments:

FIPS

For FIPS-certified RSA encryption, specify `FIPS=Y`. FIPS-certified RSA encryption uses a separate library, but is compatible with non-certified RSA encryption.

For a list of FIPS-certified components, see http://www.sybase.com/detail?id=1061806.

The algorithm must match the encryption used to create your certificates.

All strong encryption technologies are subject to export regulations.

FIPS-certified encryption requires a separate license.

server-identity-filename

The path and file name of the server identity certificate. If you are using FIPS-certified RSA encryption, you must generate your certificates using the RSA algorithm.

password

The password for the server private key. You specify this password when you create the server certificate.

Applies to

NONE and SIMPLE apply to all servers and operating systems.

TLS applies to all servers and operating systems.

For information about supporting FIPS-certified encryption, see http://www.sybase.com/detail?id=1061806.

Remarks

TDS packets are not encrypted.

You can use this option to secure communication packets between client applications and the database server using transport layer security.

The -ec option instructs the database server to accept only connections that are encrypted using one of the specified types. You must specify at least one of the supported parameters in a comma-separated list. Connections over the TDS protocol, which include Java applications using jConnect, are always accepted and are never encrypted, regardless of the usage of the -ec option. Setting the TDS protocol option to NO disallows these unencrypted TDS connections.

By default, communication packets aren't encrypted, which poses a potential security risk. If you are concerned about the security of network packets, use the -ec option. Encryption affects performance only marginally.

If the database server accepts simple encryption, but does not accept unencrypted connections, then any non-TDS connection attempts using no encryption automatically use simple encryption.

Starting the database server with `-ec SIMPLE` tells the database server to only accept connections using simple encryption. TLS connections (RSA and FIPS-certified RSA encryption) fail, and connections requesting no encryption use simple encryption.

If you want the database server to accept encrypted connections over TCP/IP, but also want to connect to the database from the local computer over shared memory, you can specify the -es option with the -ec option when starting the database server.

The `dbrsa16.dll` file contains the RSA code used for encryption and decryption. The file `dbfips16.dll` contains the code for the FIPS-certified RSA algorithm. When you connect to the database server, if the appropriate file cannot be found, or if an error occurs, a message appears in the database server messages window. The server doesn't start if the specified types of encryption cannot be initiated.

The client's and the server's encryption settings must match or the connection fails except in the following cases:

- If -ec SIMPLE is specified on the database server, but -ec NONE is not, then connections that do not request encryption can connect and automatically use simple encryption.
- If the database server specifies RSA and the client specifies FIPS-certified encryption, or vice versa, the connection succeeds. In these cases, the Encryption connection property returns the value specified by the database server.

Example

The following example specifies that connections with no encryption and simple encryption are allowed.

```
start_iq -ec NONE,SIMPLE -x tcpip c:\mydemo.db
```

The following example starts a database server that uses the RSA server certificate `rsaserver.id`.

```
start_iq -ec TLS(IDENTITY=rsaserver.id;IDENTITY_PASSWORD=test) -x tcpip c:
\mydemo.db
```

The following example starts a database server that uses the FIPS-approved RSA server certificate `rsaserver.id`.

```
start_iq -ec TLS(FIPS=Y;IDENTITY=rsaserver.id;IDENTITY_PASSWORD=test) -x tcpip c:
\mydemo.db
```

5.2 -es database server option

Allows unencrypted connections over shared memory.

Syntax

```
start_iq -ec <encryption-options> -es ...
```

Applies to

All servers and operating systems.

Remarks

This option is only effective when specified with the -ec option. The -es option instructs the database server to allow unencrypted connections over shared memory. Connections over TCP/IP must use an encryption type specified by the -ec option. This option is useful in situations where you want remote clients to use encrypted connections, but for performance reasons you also want to access the database from the local computer with an unencrypted connection.

Example

The following example specifies that connections with simple encryption and unencrypted connections over shared memory are allowed.

```
start_iq -ec SIMPLE -es -x tcpip c:\mydemo.db
```

5.3 TDS Communication Parameter

Controls whether the server allows TDS connections.

Usage

TCP/IP, NamedPipes (server side only)

Values

`YES`, `NO`

Default

`YES`

Description

To disallow TDS connections to a database server, set TDS to NO. To ensure that only encrypted connections are made to your server, these port options are the only way to disallow TDS connections.

Example

The following command starts a database server that uses the TCP/IP protocol, but disallows connections from Open Client or jConnect applications.

```
start_iq -x tcpip(TDS=NO) ...
```

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

www.sap.com/contactsap

© 2020 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see https://www.sap.com/about/legal/trademark.html for additional trademark information and notices.

THE BEST RUN SAP