# 56
# DBMS_CREDENTIAL

The DBMS_CREDENTIAL package provides an interface for authenticating and impersonating EXTPROC callout functions, as well as external jobs, remote jobs and file watchers from the SCHEDULER.

> **See Also:**
>
> - *Oracle Database Administrator's Guide* regarding Specifying Job Credentials
> - *Oracle Database Security Guide* regarding Guidelines for Securing External Processes

This chapter contains the following topics:



## DBMS_CREDENTIAL Overview

Credentials are database objects that hold a username/password pair for authenticating and impersonating EXTPROC callout functions, as well as remote jobs, external jobs and file watchers from the SCHEDULER.

They are created using the CREATE_CREDENTIAL Procedure. The procedure also allows you to specify the Windows domain for remote external jobs executed against a Windows server.

## DBMS_CREDENTIAL Security Model

Every Oracle credential has a unique credential name and you can associate a credential through its unique credential name with EXTPROC by means of a PL/SQL alias library.

In order to associate a credential with a PL/SQL alias library and external procedure, you must have the CREATE AND/OR REPLACE LIBRARY privilege or CREATE AND/OR REPLACE FUNCTION / PROCEDURE privilege and read permission of the DLL or shared object that the alias library to be associated with so that you can create and/or replace function or procedure to make use of the alias library.

Once authenticated, EXTPROC must act on behalf of the client based on client's identity defined in the supplied user credential. If not authenticated, EXTPROC must return an error message.

In order to create or alter a credential, you must have the CREATE CREDENTIAL privilege. If you are attempting to create or alter a credential in a schema other than your own, you must have the CREATE ANY CREDENTIAL privilege.

# DBMS_CREDENTIAL Operational Notes

As the existing `CREATE OR REPLACE LIBRARY` statement and `CREATE OR REPLACE FUNCTION/PROCEDURE` do not support a `CREDENTIAL` clause, this model requires syntax and semantic changes in `CREATE OR REPLACE LIBRARY` and `CREATE OR REPLACE FUNCTION/PROCEDURE` statement.

For example:

```
CREATE OR REPLACE LIBRARY test
  AS '$ORACLE_HOME/bin/test.so' CREDENTIAL ricky_cred;
CREATE OR REPLACE FUNCTION ftest1
  (x VARCHAR2, y BINARY_INTEGER)
RETURN BINARY_INTEGER
AS LANGUAGE C
LIBRARY test
NAME "negative"
PARAMETERS(x STRING, y INT);
```

The credential name defined in the `CREDENTIAL` clause is a name of a database object. Therefore, do not enclose the credential name with single or double quotes.

An example of a credential being used on an external job:

```
BEGIN
   DBMS_SCHEDULER.CREATE_JOB(
   job_name            => 'example_job',
   job_type            => 'EXECUTABLE',
   job_action          => '/bin/ls',
   credential_name     => 'ricky_cred');
END;
/
```

# Summary of DBMS_CREDENTIAL Subprograms

This table lists the `DBMS_CREDENTIAL` subprograms and briefly describes them.

**Table 56-1 DBMS_CREDENTIAL Package Subprograms**

| Subprogram | Description |
|---|---|
| CREATE_CREDENTIAL Procedure | Creates a stored username/password pair in a database object called an Oracle credential |
| DISABLE_CREDENTIAL Procedure | Disables an Oracle credential |
| DROP_CREDENTIAL Procedure | Drops an Oracle credential |
| ENABLE_CREDENTIAL Procedure | Enables an Oracle credential |
| UPDATE_CREDENTIAL Procedure | Updates an existing Oracle credential |

# CREATE_CREDENTIAL Procedure

This procedure creates a stored username/password pair in a database object called an Oracle credential. You can also use this procedure to manage the credentials used for accessing files stored in cloud object storage.

### Syntax

```
DBMS_CREDENTIAL.CREATE_CREDENTIAL (
   credential_name   IN  VARCHAR2,
   username          IN  VARCHAR2,
   password          IN  VARCHAR2,
   database_role     IN  VARCHAR2 DEFAULT NULL
   windows_domain    IN  VARCHAR2 DEFAULT NULL,
   comments          IN  VARCHAR2 DEFAULT NULL,
   enabled           IN  BOOLEAN DEFAULT TRUE);
```

### Parameters

**Table 56-2 *CREATE_CREDENTIAL Procedure Parameters***

| Parameter | Description |
|---|---|
| credential_name | Name of the credential. It can optionally be prefixed with a schema. This cannot be set to NULL. It is converted to upper case unless enclosed in double quotes. |
| username | User name to login to the operating system or remote database to run a job if this credential is chosen. This cannot be set to NULL. |
| password | Password to login to the remote operating system to run a job if this credential is chosen. It is case sensitive. |
| database_role | Whether a database job using this credential should attempt to log in with administrative privileges. Values: SYSDBA, SYSDG, SYSADMIN or SYSBACKUP. |
| windows_domain | For a Windows remote executable target, this is the domain that the specified user belongs to. The domain will be converted to uppercase automatically. |
| comments | A text string that can be used to describe the credential to the user. The Scheduler does not use this field. |
| enabled | Determines whether the credential is enabled or not |

### Usage Notes

- Credentials reside in a particular schema and can be created by any user with the CREATE CREDENTIAL or CREATE ANY CREDENTIAL system privilege. To create a credential in a schema other than your own, you must have the CREATE CREDENTIAL or CREATE ANY CREDENTIAL privilege.
- The user name is case sensitive. It cannot contain double quotes or spaces.
- Attempting to create a credential with an existing credential name returns an error. To alter an existing credential, users must drop the existing credential first using the DROP_CREDENTIAL Procedure.
- Attempting to drop an existing credential, which is already referenced by alias libraries, returns an error. To drop an existing credential without any checking, users must set the force parameter of DROP_CREDENTIAL Procedure to TRUE.

- You may also alter a credential, by means of the UPDATE_CREDENTIAL Procedure.

### Examples

#### Create a Basic Credential

```
CONN scott
Enter password: password

BEGIN
-- Basic credential.
   DBMS_CREDENTIAL.CREATE_CREDENTIAL(
      credential_name  =>  'JAMES_SMITH',
      username         =>  'james_smith',
      password         =>  'password');
END
```

#### Create a Windows Credential

```
CONN scott
Enter password: password

-- Credential including Windows domain
BEGIN
  DBMS_CREDENTIAL.CREATE_CREDENTIAL(
     credential_name   =>  'JAMES_SMITH_WIN_CREDENTIAL',
     username          =>  'james_smith',
     password          =>  'password',
     windows_domain    =>  'localdomain');
END
```

#### Display Information about Credentials

Information about credentials is displayed using the `[DBA|ALL|USER] _CREDENTIALS` views.

```
COLUMN credential_name FORMAT A25
COLUMN username FORMAT A20
COLUMN windows_domain FORMAT A20
SELECT credential_name,
       username,
       windows_domain
FROM   user_credentials
ORDER BY credential_name;

CREDENTIAL_NAME           USERNAME             WINDOWS_DOMAIN
------------------------- -------------------- --------------------
JAMES_SMITH_CREDENTIAL       james_smith
JAMES_SMITH_WIN_CREDENTIAL   james_smith             LOCALDOMAIN
2 rows selected.

SQL>
```

## DISABLE_CREDENTIAL Procedure

This procedure disables an Oracle credential.

### Syntax

```
DBMS_CREDENTIAL.DISABLE_CREDENTIAL (
   credential_name   IN  VARCHAR2,
   force             IN  BOOLEAN DEFAULT FALSE);
```

### Parameters

**Table 56-3 *DISABLE_CREDENTIAL Procedure Parameters***

| Parameter | Description |
|---|---|
| credential_name | Name of the credential. It can optionally be prefixed with a schema. This cannot be set to NULL. It is converted to upper case unless enclosed in double quotes. |
| force | If FALSE, the credential is not disabled provided it has no dependency on any existing scheduler job or PL/SQL library. An error is returned if the dependency is observed.If TRUE, the credential is disabled whether or not there is any scheduler job or PL/SQL library referencing it. |

### Usage Notes

- Credentials reside in a particular schema and can be disabled by any user with the CREATE CREDENTIAL or CREATE ANY CREDENTIAL system privilege. To disable a credential in a schema other than your own, you must have the CREATE ANY CREDENTIAL privilege.
- A credential for an OS user can be viewed as an entry point into an operating system as a particular user. Allowing a credential to be disabled lets an administrator (or credential owner) to quickly, easily and reversibly disallow all logins from the database to the OS as a particular user of external jobs, database jobs, file transfers, external procedures, and file watching. To enable an existing disabled credential, you need to use the ENABLE_CREDENTIAL Procedure.
- A library can become invalid if the properties of the credential – windows domain, username, password, its enable/disable bit – are changed.

## DROP_CREDENTIAL Procedure

This procedure drops an Oracle credential.

### Syntax

```
DBMS_CREDENTIAL.DROP_CREDENTIAL (
   credential_name   IN  VARCHAR2,
   force             IN  BOOLEAN DEFAULT FALSE);
```

### Parameters

**Table 56-4 *DROP_CREDENTIAL Procedure Parameters***

| Parameter | Description |
|---|---|
| credential_name | Name of the credential. It can optionally be prefixed with a schema. This cannot be set to NULL. |
| force | If set to FALSE, the credential must not be referenced by any EXTPROC alias library or an error is raised. If set to TRUE, the credential is dropped whether or not there are extproc alias libraries referencing it. EXTPROC alias libraries that reference the dropped credential become invalid. |

### Usage Notes

Only the owner of a credential or a user with the `CREATE ANY CREDENTIAL` system privilege may drop the credential.

### Examples

```
EXEC DBMS_CREDENTIAL.DROP_CREDENTIAL('JAMES_SMITH_CREDENTIAL', FALSE);
EXEC DBMS_CREDENTIAL.DROP_CREDENTIAL('JAMES_SMITH_WIN_CREDENTIAL', FALSE);
```

## ENABLE_CREDENTIAL Procedure

This procedure enables an Oracle credential.

### Syntax

```
DBMS_CREDENTIAL.ENABLE_CREDENTIAL (
   credential_name   IN  VARCHAR2);
```

### Parameters

**Table 56-5** ***ENABLE_CREDENTIAL Procedure Parameters***

| Parameter | Description |
|---|---|
| `credential_name` | Name of the credential. It can optionally be prefixed with a schema. This cannot be set to `NULL`. It is converted to upper case unless enclosed in double quotes. |

### Usage Notes

- Credentials reside in a particular schema and can be disabled by any user with the `CREATE CREDENTIAL OR CREATE ANY CREDENTIAL` system privilege. To disable a credential in a schema other than your own, you must have the `CREATE CREDENTIAL OR CREATE ANY CREDENTIAL` privilege.
- A credential for an OS user can be viewed as an entry point into an operating system as a particular user. Allowing a credential to be disabled would allow an administrator (or credential owner) to quickly, easily and reversibly disallow all logins from the database to the OS as a particular user (external jobs, file transfers, external procedures, file watching). To disable an existing credential, you need to use the DISABLE_CREDENTIAL Procedure.
- A library can become invalid if the properties of the credential – windows domain, username, password, its enable/disable bit – are changed.

## UPDATE_CREDENTIAL Procedure

This procedure updates an existing Oracle credential.

### Syntax

```
DBMS_CREDENTIAL.UPDATE_CREDENTIAL (
   credential_name   IN   VARCHAR2,
   attribute         IN   VARCHAR2,
   value             IN   VARCHAR2);
```

### Parameters

**Table 56-6 *UPDATE_CREDENTIAL Procedure Parameters***

| Parameter | Description |
|---|---|
| credential_name | Name of the credential. It can optionally be prefixed with a schema. This cannot be set to NULL. It is converted to upper case unless enclosed in double quotation marks. |
| attribute | Name of attribute to update: USERNAME, PASSWORD, WINDOWS_DOMAIN, DATABASE_ROLE or COMMENTS |
| value | New value for the selected attribute |

### Usage Notes

- Credentials reside in a particular schema and can be created by any user with the CREATE CREDENTIAL or CREATE ANY CREDENTIAL system privilege. To create a credential in a schema other than your own, you must have the CREATE ANY CREDENTIAL privilege.
- The user name is case sensitive. It cannot contain double quotes or spaces.
- EXTPROC alias libraries that reference the updated credential will become invalid. A library becomes invalid if the properties of the credential – windows domain, username, password, its enable/disable bit – are changed.

### Examples

#### Update a Basic Credential

```
CONN scott
Enter password: password

BEGIN
-- Basic credential.
   DBMS_CREDENTIAL.UPDATE_CREDENTIAL (
      credential_name  =>  'JAMES_SMITH_CREDENTIAL',
      attribute        =>  'password',
      value            =>  'password2');

   DBMS_CREDENTIAL.UPDATE_CREDENTIAL (
      credential_name  =>  'JAMES_SMITH_CREDENTIAL',
      attribute        =>  'username',
      value            => 'james_smith');
END;
```

#### Update a Windows Credential

```
CONN scott
Enter password: password

-- Credential including Windows domain
BEGIN
  DBMS_CREDENTIAL.UPDATE_CREDENTIAL(
     credential_name   =>  'JAMES_SMITH_WIN_CREDENTIAL',
     username          =>  'james_smith',
     password          =>  'password',
     windows_domain    =>  'localdomain');
END
```

### Display Information about Credentials

Information about credentials is displayed using the `[DBA|ALL|USER] _CREDENTIALS` views.

```
COLUMN credential_name FORMAT A25
COLUMN username FORMAT A20
COLUMN windows_domain FORMAT A20
SELECT credential_name,
       username,
       windows_domain
FROM   all_credentials
ORDER BY credential_name;

CREDENTIAL_NAME           USERNAME             WINDOWS_DOMAIN
------------------------- -------------------- --------------------
JAMES_SMITH_CREDENTIAL       james_smith
JAMES_SMITH_WIN_CREDENTIAL   james_smith             LOCALDOMAIN
2 rows selected.

SQL>
```