208
DBMS_TSDP_PROTECT

The DBMS_TSDP_PROTECT package provides an interface to configure transparent sensitive data protection (TSDP) policies in conjunction with the DBMS_TSDP_MANAGE package.

DBMS_TSDP_PROTECT is available with the Enterprise Edition only.

This chapter contains the following topics:

Related Topics

- DBMS_TSDP_MANAGE
  The DBMS_TSDP_MANAGE package provides an interface to import and manage sensitive columns and sensitive column types in the database, and is used in conjunction with the DBMS_TSDP_PROTECT package with regard to transparent sensitive data protection (TSDP) policies.

> **See Also:**
>
> *Oracle Database Security Guide*

DBMS_TSDP_PROTECT Overview

Use the DBMS_TSDP_PROTECT package to create transparent sensitive data protection policies, configure protection by associating the policies with sensitive types, and to enable and disable the configured protection.

Sensitive types can be added using the DBMS_TSDP_MANAGE package.

DBMS_TSDP_PROTECT Security Model

All procedures are executed with invoker's rights. Typically, a security administrator should have the EXECUTE privilege for this package.

DBMS_TSDP_PROTECT Constants

DBMS_TSDP_PROTECT defines the TSDP_PARAM_MAX constant for use when specifying parameter values.

This constant is described in the following table.

Table 208-1 DBMS_TDSP_PROTECT Constants - Compression Types

Constant	Type	Value	Description
TSDP_PARAM_MAX	INTEGER	4000	Maximum length of the parameter value that can be specified in FEATURE_OPTIONS

DBMS_TSDP_PROTECT Data Structures

The DBMS_TSDP_PROTECT package defines two TABLE types.

Table Types

- FEATURE_OPTIONS Table Type

  POLICY_CONDITIONS Table Type

FEATURE_OPTIONS Table Type

The following type is an associative array of VARCHAR2(TSDP_PARAM_MAX) that is indexed by VARCHAR2(M_IDEN).

Syntax

```
TYPE FEATURE_OPTIONS IS TABLE OF VARCHAR2(TSDP_PARAM_MAX)
INDEX BY VARCHAR2(M_IDEN);
```

POLICY_CONDITIONS Table Type

The following type is an associative array of VARCHAR2(TSDP_PARAM_MAX) that is indexed by PLS_INTEGER.

Syntax

```
TYPE POLICY_CONDITIONS IS TABLE OF VARCHAR2(TSDP_PARAM_MAX)
INDEX BY PLS_INTEGER;
```

Summary of DBMS_TSDP_PROTECT Subprograms

This table lists the DBMS_TSDP_PROTECT subprograms and briefly describes them.

Table 208-2 DBMS_TSDP_PROTECT Package Subprograms

Subprogram	Description
ADD_POLICY Procedure	Creates a TSDP policy
ALTER_POLICY Procedure	Alters a TDSP policy
ASSOCIATE_POLICY Procedure	Associates or disassociates a TSDP policy with a sensitive column type
DISABLE_PROTECTION_COLUMN Procedure	Disables protection for columns
DISABLE_PROTECTION_SOURCE Procedure	Disables protection based on the source of truth for the sensitive columns
DISABLE_PROTECTION_TYPE Procedure	Disables protection for a sensitive column type
DROP_POLICY Procedure	Removes a TDSP policy
ENABLE_PROTECTION_COLUMN Procedure	Enables protection for columns
ENABLE_PROTECTION_SOURCE Procedure	Enables protection based on the source of truth for the sensitive columns
ENABLE_PROTECTION_TYPE Procedure	Enables protection for a sensitive column type

ADD_POLICY Procedure

This procedure creates a TDSP policy.

Syntax

```
DBMS_TSDP_PROTECT.ADD_POLICY (
   policy_name              IN VARCHAR2,
   security_feature         IN PLS_INTEGER,
   policy_enable_options    IN FEATURE_OPTIONS,
   policy_apply_condition   IN POLICY_CONDITION DEFAULT TSDP$default_condition);
```

Parameters

Table 208-3 *ADD_POLICY Procedure Parameters*

Parameter	Description
`policy_name`	Name of the policy being created. The maximum length for this identifier is `M_IDEN`. This follows the Oracle naming convention.
`security_feature`	Oracle security feature with which the policy is associated. Allowed values: • `DBMS_TSDP_PROTECT.REDACT` • `DBMS_TSDP_PROTECT.VPD` • `DBMS_TSDP_PROTECT.UNIFIED_AUDIT` • `DBMS_TSDP_PROTECT.FINE_GRAINED_AUDIT` • `DBMS_TSDP_PROTECT.COLUMN_ENCRYPTION`
`policy_enable_options`	Initialized with the parameter-value pairs corresponding to the `security_feature` setting

Table 208-3 (Cont.) *ADD_POLICY Procedure Parameters*

Parameter	Description
policy_apply_condition	Initialized with the property-value pairs that must be satisfied in order to apply the corresponding policy_enable_options. This is an associative array with Property as the key (PLS_INTEGER). Example: example_policy_condition(*Property*)= *property_value*. Permissible values for Property: • DBMS_TSDP_PROPERTY.DATATYPE • DBMS_TSDP_PROPERTY.LENGTH • DBMS_TSDP_PROPERTY.PARENT_SCHEMA • DBMS_TSDP_PROPERTY.PARENT_TABLE

Usage Notes

To create the TDSP policy, you must include the procedure in an anonymous block that defines the type of security feature that will use the policy and conditions to test when the policy is enabled. For more information, see *Oracle Database Security Guide*.

Examples

Create a policy PARTIAL_MASK_POLICY:

```
DECLARE
  redact_feature_options DBMS_TSDP_PROTECT.FEATURE_OPTIONS;
  policy_conditions DBMS_TSDP_PROTECT.POLICY_CONDITIONS;
BEGIN
  redact_feature_options ('expression') :=
   'SYS_CONTEXT(''USERENV'',''SESSION_USER'')      ='''APPUSER''';
  redact_feature_options ('function_type')        := 'DBMS_REDACT.PARTIAL';
  redact_feature_options ('function_parameters')  := 'STR, VVVVVVVVV,VVVVVVVVV, *,  1,
6';
  policy_conditions(DBMS_TSDP_PROTECT.DATATYPE)   := 'VARCHAR2';
 DBMS_TSDP_PROTECT.ADD_POLICY
  ('PARTIAL_MASK_POLICY',  DBMS_TSDP_PROTECT.REDACT, redact_feature_options,
policy_conditions);
END;
```

ALTER_POLICY Procedure

This procedure alters an existing TDSP policy

Syntax

```
DBMS_TSDP_PROTECT.ALTER_POLICY (
   policy_name              IN VARCHAR2,
   policy_enable_options    IN FEATURE_OPTIONS,
   policy_apply_condition   IN POLICY_CONDITION default TSDP$default_condition);
```

Parameters

Table 208-4 *ALTER_POLICY Procedure Parameters*

Parameter	Description
policy_name	Name of the policy to alter

Table 208-4 (Cont.) *ALTER_POLICY Procedure Parameters*

Parameter	Description
`policy_enable_options`	Initialized with the parameter-value pairs corresponding to the security feature
`policy_apply_condition`	Initialized with the property-value pairs that must be satisfied in order to apply the corresponding `policy_enable_options`. This is an associative array with Property as the key (`PLS_INTEGER`). Example: `example_policy_condition(`*`Property`*`)=`*`property_value`*. Permissible values for Property: • `DBMS_TSDP_PROPERTY.DATATYPE` • `DBMS_TSDP_PROPERTY.LENGTH` • `DBMS_TSDP_PROPERTY.PARENT_SCHEMA` • `DBMS_TSDP_PROPERTY.PARENT_TABLE`

Usage Notes

- If the `policy_apply_condition` matches an existing condition for the policy, then the corresponding enable options are updated with `policy_enable_options`.
- If the `policy_apply_condition` does not match any existing condition for the policy, the combination of `policy_enable_options` and `policy_apply_condition` is added to the policy.

Examples

Add a new combination of `policy_apply_condition` and `policy_enable_options` to an existing policy `PARTIAL_MASK_POLICY`:

```
DECLARE
  redact_feature_options DBMS_TSDP_PROTECT.FEATURE_OPTIONS;
  policy_conditions DBMS_TSDP_PROTECT.POLICY_CONDITIONS;
BEGIN
   redact_feature_options ('expression') :=
       'SYS_CONTEXT(''USERENV'',''SESSION_USER'')=''APPUSER''';
   redact_feature_options ('function_type') := 'DBMS_REDACT.PARTIAL';
   redact_feature_options ('function_parameters') := 'STR, VVVVVVVVV,VVVVVVVVV, *,
        1, 6';
   policy_conditions (DBMS_TSDP_PROTECT.DATATYPE) := 'VARCHAR2';
   DBMS_TSDP_PROTECT.ALTER_POLICY ('PARTIAL_MASK_POLICY',  redact_feature_options,
policy_conditions);
END;
```

ASSOCIATE_POLICY Procedure

This procedure associates or disassociates a TSDP policy with a sensitive column type.

Syntax

```
DBMS_TSDP_PROTECT.ASSOCIATE_POLICY (
   policy_name        IN  VARCHAR2,
   sensitive_type     IN  VARCHAR2,
   associate          IN  BOOLEAN DEFAULT TRUE);
```

Parameters

Table 208-5 *ASSOCIATE_POLICY Procedure Parameters*

Parameter	Description
policy_name	Name of the TDSP policy
sensitive_type	Name of the sensitive column type:
associate	Associate or Disassociate. TRUE implies Associate

Usage Notes

Both the policy and the sensitive column type should exist in the database.

Examples

Associate `PARTIAL_MASK_POLICY` with `SSN_TYPE`:

```
DBMS_TSDP_PROTECT.ASSOCIATE_POLICY ('PARTIAL_MASK_POLICY', 'SSN_TYPE');
```

DISABLE_PROTECTION_COLUMN Procedure

This procedure disables protection for columns.

Syntax

```
DBMS_TSDP_PROTECT.DISABLE_PROTECTION_COLUMN (
   schema_name        IN  VARCHAR2 DEFAULT '%',
   table_name         IN  VARCHAR2 DEFAULT '%',
   column_name        IN  VARCHAR2 DEFAULT '%',
   policy_name        IN  VARCHAR2 DEFAULT NULL);
```

Parameters

Table 208-6 *DISABLE_PROTECTION_COLUMN Procedure Parameters*

Parameter	Description
schema_name	Name of the schema containing the column
table_name	Table containing the column
column_name	Column name
policy_name	Optional policy name. If given, only this policy is disabled.

Examples

Disable TSDP policies associated with the corresponding sensitive column types for columns that reside in schema with name like `%PAYROLL%`, table name like `EMP%`, and column name like `SAL%`:

```
EXEC DBMS_TSDP_PROTECT.DISABLE_PROTECTION_COLUMN ('%PAYROLL%', 'EMP%', 'SAL%');
```

DISABLE_PROTECTION_SOURCE Procedure

This procedure disables protection based on the source of truth for the sensitive columns.

Syntax

```
DBMS_TSDP_PROTECT.DISABLE_PROTECTION_SOURCE (
   discovery_sourcename      IN  VARCHAR2);
```

Parameters

Table 208-7 *DISABLE_PROTECTION_SOURCE Procedure Parameters*

Parameter	Description
discovery_sourcename	Name of the discovery source. This could be the Application Data Model (ADM) name or the database user.

Examples

Disable protection for all columns corresponding to `ADM_Demo`:

```
DBMS_TSDP_PROTECT.DISABLE_PROTECTION_SOURCE ('ADM_Demo');
```

DISABLE_PROTECTION_TYPE Procedure

This procedure disables protection for a sensitive column type.

Syntax

```
DBMS_TSDP_PROTECT.DISABLE_PROTECTION_TYPE (
   sensitive_type         IN VARCHAR2);
```

Parameters

Table 208-8 *DISABLE_PROTECTION_TYPE Procedure Parameters*

Parameter	Description
sensitive_type	Name of the sensitive column type

Examples

Disable protection for all columns identified by `SSN_TYPE`:

```
DBMS_TSDP_PROTECT.DISABLE_PROTECTION_TYPE ('SSN_TYPE');
```

DROP_POLICY Procedure

This procedure removes a TDSP policy or one of its condition-enable_options combinations.

Syntax

```
DBMS_TSDP_PROTECT.DROP_POLICY (
   policy_name              IN VARCHAR2,
   policy_apply_condition   IN POLICY_CONDITIONS);
```

```
DBMS_TSDP_PROTECT.DROP_POLICY (
   policy_name              IN VARCHAR2);
```

Parameters

Table 208-9 *DROP_POLICY Procedure Parameters*

Parameter	Description
policy_name	Name of the policy to drop
policy_apply_condition	To be initialized with the relevant condition

Usage Notes

- The combination of policy_conditions and policy_enable_options can be dropped from a TSDP policy by giving the policy_apply_condition parameter. The default condition-default options combination can also be dropped (if it exists for the policy) by passing an empty associative array of type DBMS_TSDP_PROTECT.POLICY_CONDITION.
- If the condition-enable_options combination that is being dropped is the last condition-enable_options combination for the policy, the policy itself is dropped.
- A policy can be completely dropped by using the overloaded of the procedure that takes only policy_name.
- A policy or one of its conditions can be dropped only if the policy is not associated with any sensitive column type. This also means that a policy that is being dropped is not enabled on any column (object).

Examples

Dropping the condition-enable_options combination based on a specific condition:

```
DECLARE
   policy_conditions DBMS_TSDP_PROTECT.POLICY_CONDITIONS;
BEGIN
   policy_conditions (DBMS_TSDP_PROTECT.DATATYPE) := 'VARCHAR2';
   DBMS_TSDP_PROTECT.DROP_POLICY ('PARTIAL_MASK_POLICY', policy_conditions);
END;
```

The default condition-enable_options combination can be dropped by passing an empty associative array of type DBMS_TSDP_PROTECT.POLICY_CONDITIONS for the policy_apply_condition parameter:

```
DECLARE
   policy_conditions DBMS_TSDP_PROTECT.POLICY_CONDITIONS;
BEGIN
  DBMS_TSDP_PROTECT.DROP_POLICY ('redact_partial_cc', policy_conditions);
END;
```

Dropping a TSDP policy:

```
BEGIN
   DBMS_TSDP_PROTECT.DROP_POLICY(
   policy_name => 'PARTIAL_MASK_POLICY');
END;
```

ENABLE_PROTECTION_COLUMN Procedure

This procedure enables protection for columns.

Syntax

```
DBMS_TSDP_PROTECT.ENABLE_PROTECTION_COLUMN (
   schema_name         IN  VARCHAR2 DEFAULT '%',
   table_name          IN  VARCHAR2 DEFAULT '%',
   column_name         IN  VARCHAR2 DEFAULT '%',
   policy_name         IN  VARCHAR2 DEFAULT NULL);
```

Parameters

Table 208-10 *ENABLE_PROTECTION_COLUMN Procedure Parameters*

Parameter	Description
schema_name	Name of the schema containing the column
table_name	Table containing the column
column_name	Column name
policy_name	Optional policy name. If given, only this policy is enabled.

Usage Notes

- Only a TSDP Policy that is associated with the sensitive column type of the sensitive column can be enabled using this Procedure.
- LIKE condition is used for schema_name, table_name and column_name. AND semantics is followed.

Examples

Enable TSDP policies associated with the corresponding sensitive column types for columns that reside in schema with name like %PAYROLL%, table name like EMP%, and column name like SAL%:

```
DBMS_TSDP_PROTECT.ENABLE_PROTECTION_COLUMN ('%PAYROLL%', 'EMP%', 'SAL%');
```

ENABLE_PROTECTION_SOURCE Procedure

This procedure enables protection based on the source of truth for the sensitive columns.

Syntax

```
DBMS_TSDP_PROTECT.ENABLE_PROTECTION_SOURCE (
   discovery_sourcename      IN  VARCHAR2);
```

Parameters

Table 208-11 *ENABLE_PROTECTION_SOURCE Procedure Parameters*

Parameter	Description
discovery_sourcename	Name of the discovery source. This could be the Application Data Model (ADM) name or the database user.

Examples

Enable protection for all columns corresponding to ADM_Demo:

```
DBMS_TSDP_PROTECT.ENABLE_PROTECTION_SOURCE ('ADM_Demo');
```

ENABLE_PROTECTION_TYPE Procedure

This procedure enables protection for a sensitive column type.

Syntax

```
DBMS_TSDP_PROTECT.ENABLE_PROTECTION_TYPE (
   sensitive_type         IN VARCHAR2);
```

Parameters

Table 208-12 *ENABLE_PROTECTION_TYPE Procedure Parameters*

Parameter	Description
sensitive_type	Name of the sensitive column type

Examples

Enable protection for all columns identified by SSN_TYPE:

```
DBMS_TSDP_PROTECT.ENABLE_PROTECTION_TYPE ('SSN_TYPE');
```