# DBMS\_SODA

The <code>DBMS\_SODA</code> package is a <code>PL/SQL</code> package implementing Simple Oracle Document Access (<code>SODA</code>). <code>SODA</code> allows you to use the Oracle Database as a <code>NoSQL</code> document store. The core abstraction provided by <code>SODA</code> is that of document collections. The <code>DBMS\_SODA</code> package allows you to create, list, and delete document collections from <code>PL/SQL</code>, and to perform CRUD (create, replace, update, delete) operations on documents. All <code>DDL</code> functions are encapsulated within this package.

This chapter contains the following topics:

- DBMS\_SODA Security Model
- Summary of DBMS\_SODA Subprograms
- Summary of SODA Online Redefinition Subprograms

### See Also:

- Oracle Database SODA for PL/SQL Developer's Guide
- SODA Types

# **DBMS SODA Security Model**

This package is available to users with the  ${\tt SODA\_APP}$  role.

All SODA types (packages and types) are SYS types. PUBLIC is granted EXECUTE privilege on the DBMS SODA described in this chapter.

# Summary of DBMS\_SODA Subprograms

This table lists the DBMS SODA subprograms in alphabetical order and briefly describes them.

Table 182-1 DBMS\_SODA Package Subprograms

Purpose
Creates a collection using the collection name and metadata.
Drops an existing collection from the user's schema. This also removes all the documents in the collection.
This function returns the current database SCN number.
This function returns the current database timestamp value.

Table 182-1 (Cont.) DBMS\_SODA Package Subprograms

Subprogram	Purpose
GET_DEFAULT_METADATA_AS_CLOB Function	This function returns the default metadata.
GET_DEFAULT_METADATA_AS_VARCHAR2 Function	This function returns the default metadata.
LIST_COLLECTION_NAMES Function	Lists the collection names in the user's schema as a table of NVARCHAR2.
OPEN_COLLECTION Function	Opens an existing collection.

# CREATE\_COLLECTION Function

Creates a collection using the collection name and metadata. Uses the settings specified in the metadata and auto-assigns the ones that are not, and returns the collection object. If the metadata argument is omitted or set to <code>NULL</code>, a collection is created with default metadata. The returned collection is open for <code>read</code> and/or <code>write</code> operations. If a collection already exists, the function just opens and returns the collection object.

### **Syntax**

### **Parameters**

Table 182-2 CREATE\_COLLECTION Parameters

Parameter	Description
collection_Name	The name of the collection.
	The value of collection_Name is case-sensitive.
metadata	The metadata of the collection in VARCHAR2 format.
create_Mode	<ul> <li>Valid values are:         <ul> <li>DBMS_SODA.CREATE_MODE_DDL (default). First attempts to create a new table for the collection. If table exists already, tries to map that existing table to the collection. Minimal checking is performed to ensure the table shape matches supplied collection metadata (if not, then an error is returned).</li> </ul> </li> <li>DBMS_SODA.CREATE_MODE_MAP. Tries to map an existing table to the collection. Minimal checking is performed to ensure the table shape matches supplied collection metadata (if not, then an error is returned).</li> </ul>

### **Return Values**

The function returns a Soda\_Collection\_T object representing the collection.

### **Exceptions**

- Descriptor Error—if the input descriptor is invalid
- Error—if an error occurs while creating the collection



Oracle Database SODA for PL/SQL Developer's Guide

# DROP\_COLLECTION Function

Drops an existing collection from the user's schema. This also removes all the documents in the collection.

### **Syntax**

### **Parameters**

### Table 182-3 DROP\_COLLECTION Parameters

Parameter	Description
collection_Name	The name of the collection.
	The value of collection_Name is case-sensitive.
purge	The default value is FALSE.
drop_Mapped_Table	The default value is FALSE.

### Return values

This function returns the following values:

- 1—if the collection was dropped successfully
- 0—if the collection does not exist

### **Exceptions**

If an error occurs while dropping the collection, for example, due to uncommitted writes to the collection or privilege issues.



Oracle Database SODA for PL/SQL Developer's Guide



# GET\_AS\_OF\_SCN Function

This function returns the current database SCN number.

### **Syntax**

```
DBMS_SODA.GET_AS_OF_SCN ( )
RETURN NUMBER;
```

### Return values

This function returns the current database SCN number.

## GET\_AS\_OF\_TIMESTAMP Function

This function returns the current database timestamp value.

### **Syntax**

```
DBMS_SODA.GET_AS_OF_TIMESTAMP ( )
RETURN VARCHAR2;
```

### Return values

This function returns the current database timestamp value.

## GET\_DEFAULT\_METADATA\_AS\_CLOB Function

This function returns the default metadata.

### **Syntax**

```
DBMS_SODA.GET_DEFAULT_METADATA_AS_CLOB ( )
RETURN CLOB;
```

### **Return Values**

This function returns the default metadata using the CLOB datatype.

# GET\_DEFAULT\_METADATA\_AS\_VARCHAR2 Function

This function returns the default metadata.

### **Syntax**

```
DBMS_SODA.GET_DEFAULT_METADATA_AS_VARCHAR2 ( )
RETURN VARCHAR2;
```

### **Return Values**

This function returns the default metadata using the VARCHAR2 datatype.

## LIST COLLECTION NAMES Function

This function returns a list of collection names in the user's schema as a table of NVARCHAR2.

### **Syntax**

```
DBMS_SODA.LIST_COLLECTION_NAMES ()
  RETURN SODA_CollName_List_T;
```

### **Return Values**

This function returns a list of collection names as a table of NVARCHAR2 (255). The collection list is empty if there are no collections in the schema.

### **Exceptions**

Error—if an error occurs while listing the collection names.



Oracle Database SODA for PL/SQL Developer's Guide

# **OPEN\_COLLECTION** Function

Opens an existing collection for read and/or write operations.

### **Syntax**

### **Parameters**

### Table 182-4 OPEN\_COLLECTION Parameters

Parameter	Description
collection_Name	The name of the collection.
	The value of collection_Name is case-sensitive.

### **Return Values**

This function returns the following values:

- a collection object which is open
- NULL, if the collection does not exist

### **Exceptions**

Error—if an error occurs while creating the collection

### See Also:

- · Opening an Existing Document Collection with SODA for PL/SQL
- Checking Whether a Given Collection Exists with SODA for PL/SQL

# Summary of SODA Online Redefinition Subprograms

This table lists the SODA Online Redefinition subprograms in alphabetical order and briefly describes them.

Table 182-5 SODA Online Redefinition Subprograms

Subprogram	Purpose
ABORT_REDEF_COLLECTION Procedure	This procedure reverts the changes made to a collection if there are any errors in CREATE_INTERIM_COLLECTION, START_REDEF_COLLECTION, COPY_COLLECTION_DEPENDENTS, SYNC_INTERIM_COLLETION, or due to any other reasons.
CAN_REDEF_COLLECTION Procedure	This procedure checks if the data table can be redefined. If the data table cannot be redefined, errors are raised.
COPY_COLLECTION_DEPENDENTS Procedure	This procedure copies all the dependents that are defined on the original table to the interim table. However, if the interim table is JSON type, constraints like IS JSON NOT NULL will not be copied because those are extraneous.
CREATE_INTERIM_COLLECTION Procedure	This procedure creates an interim SODA collection. The interim_metadata specifies the changes the user needs to make. For example, a delta applied as a patch to the original matadata.
FINISH_REDEF_COLLECTION Procedure	This procedure performs the following tasks in one atomic transaction. It introduces a downtime. This can only be called after every dependent required for a SODA data table is present on the interim collection, otherwise an error is raised.
START_REDEF_COLLECTION Procedure	This procedure starts the redefinition process. Copies the existing data from the original table to the interim table and performs the required transformations on the redefined columns.
SYNC_INTERIM_COLLETION Procedure	This procedure syncronizes the changes to the interim table if the original data table was changed This procedure can only be called after every dependent required for the SODA data table is present on the interim collection, otherwise, an error is raised.
TO_UUID Function	This function is used to remap the version column of the collection table during Data Pump Import to UUID values. This function is only for use with the REMAP_DATA feature of Data Pump.



# ABORT\_REDEF\_COLLECTION Procedure

This procedure reverts the changes made to a collection if there are any errors in CREATE\_INTERIM\_COLLECTION, START\_REDEF\_COLLECTION, COPY\_COLLECTION\_DEPENDENTS, SYNC INTERIM COLLETION, or due to any other reasons.

### **Syntax**

#### **Parameters**

### Table 182-6 ABORT\_REDEF\_COLLECTION Procedure Parameters

Parameter	Description
collection_name	The name of the collection to be redefined.
interim_collection_name	The name used for the interim collection.

## CAN\_REDEF\_COLLECTION Procedure

This procedure checks if the data table can be redefined. If the data table cannot be redefined, errors are raised.

### **Syntax**

```
DBMS_SODA.CAN_REDEF_COLLECTION (
    collection_name IN NVARCHAR2);
```

### **Parameters**

### Table 182-7 CAN\_REDEF\_COLLECTION Procedure Parameters

Parameter	Description
collection_name	The name of the collection to be redefined.

## COPY\_COLLECTION\_DEPENDENTS Procedure

This procedure copies all the dependents that are defined on the original table to the interim table.

### **Syntax**



#### **Parameters**

Table 182-8 COPY\_COLLECTION\_DEPENDENTS Procedure Parameters

Parameter	Description
collection_name	The name of the collection to be redefined.
interim_collection_name	The name used for the interim collection.
ignore_error	Ignore the errors encountered in the process and proceed.
num_errors	The number of errors encountered in the process.

## CREATE\_INTERIM\_COLLECTION Procedure

This procedure creates an interim <code>SODA</code> collection. The <code>interim\_metadata</code> specifies the changes the user needs to make. For example, a delta applied as a patch to the original matadata.

### **Syntax**

### **Parameters**

Table 182-9 CREATE\_INTERIM\_COLLECTION Procedure Parameters

Parameter	Description
collection name	The name of the collection to be redefined.
_	The name used for the interim collection.
interim_metadata	The metadata snippet specifying the columns that need to be redefined and how they need to be redefined.

### **Usage Notes**

The <code>interim\_metadata</code> parameter can contain the following fields. An error is raised if other fields are present and if the fields are not the same as those in the metadata for the original collection.

- Set contentColumn.sqlType to JSON in order to redefine the CONTENT column to JSON type, automatically generating the VERSION column if needed.
- Set versionColumn.method to UUID to redefine the VERSION column to use UUID.
- Set tableName to specify a name for the data table of this interim collection. If this field is not present, a default data table name will be generated according to the rules of table name defaulting outlined in Default Naming of a Collection Table. However, if the table already exists, the collection is created using the MAP mode. In this scenario, make sure that the mapped table does not have any constraints defined.
- The user can combine the above fields in the supplied <code>interim\_metadata</code> snippet. For example, the most common use case is to both redefine the content column to <code>JSON</code> type and redefine the version column to <code>UUID</code>, which can be achieved by setting the

```
interim metadata parameter to {"contentColumn" : {"sqlType": "JSON"},
"versionColumn" : {"method": "UUID"}}.
```

## FINISH REDEF COLLECTION Procedure

This procedure performs the following tasks in one atomic transaction. It introduces a downtime. This can only be called after every dependent required for a SODA data table is present on the interim collection, otherwise an error is raised.

### **Syntax**

```
DBMS_SODA.FINISH_REDEF_COLLECTION (
  collection name IN NVARCHAR,
  interim collection name IN NVARCHAR2,
  dml lock timeout IN PLS INTEGER DEFAULT NULL);
```

### **Parameters**

### Table 182-10 FINISH REDEF COLLECTION Procedure Parameters

Parameter	Description
collection_name	The name of the collection to be redefined.
interim_collection_name	The name used for the interim collection.
dml_lock_timeout	The wait period in seconds. If the lock is not acquired within this wait period, an error occurs.

### **Usage Notes**

This action cannot be undone. After this procedure is executed, you cannot perform ROLLBACK procedure provided by the DBMS REDEFINITION package.



### Caution:

Before calling FINISH REDEF COLLECTION Procedure, the interim collection should be tried out to make sure it can be accessed and is working as expected. In case of any issues, use ABORT\_REDEF\_COLLECTION Procedure.

It is very important to do this before calling FINISH REDEF COLLECTION because the latter cannot be reversed.

### START REDEF COLLECTION Procedure

This procedure starts the redefinition process. Copies the existing data from the original table to the interim table and performs the required transformations on the redefined columns.

### **Syntax**

```
DBMS SODA.START REDEF COLLECTION (
  collection_name IN NVARCHAR,
   interim_collection_name IN NVARCHAR2,
  copy_vpd_opt IN DEFAULT NULL, refresh_dep_mviews IN DEFAULT NULL);
```



#### **Parameters**

Table 182-11 START\_REDEF\_COLLECTION Procedure Parameters

Parameter	Description
collection_name	The name of the collection to be redefined.
<pre>interim_collection_name</pre>	The name used for the interim collection.
copy_vpd_opt	Can be either DBMS_REDEFINITION.CONS_VPD_MANUAL or DBMS_REDEFINITION.CONS_VPD_NONE.
	DBMS_REDEFINITION.CONS_VPD_MANUAL is used to indicate to copy VPD policies manually. DBMS_REDEFINITION.CONS_VPD_NONE is used to indicate that there are no VPD policies on the original table. If NULL, DBMS_REDEFINITION.CONS_VPD_NONE is used.
refresh_dep_mviews	Can be N or Y. When set to N, fast refresh of dependent materialized views is performed when the START_REDEF_TABLE procedure is run, each time the SYNC_INTERIM_TABLE procedure is run, and when the FINISH_REDEF_TABLE procedure is run. If NULL, N is used.

# SYNC\_INTERIM\_COLLETION Procedure

This procedure synchronizes the changes to the interim table if the original data table was changed. This procedure can only be called after every dependent required for the SODA data table is present on the interim collection, otherwise, an error is raised.

### **Syntax**

### **Parameters**

Table 182-12 SYNC\_INTERIM\_COLLETION Procedure Parameters

Parameter	Description
collection_name	The name of the collection to be redefined.
interim_collection_name	The name used for the interim collection.

# TO\_UUID Function

This function is used to remap the version column of the collection table during Data Pump Import to UUID values. This function is only for use with the REMAP DATA feature of Data Pump.

### **Syntax**



### **Parameters**

Table 182-13 TO\_UUID Parameters

Parameter	Description
placeholder	This parameter is not used and can be ignored. It is used due to the syntactic constraints for the Data Pump REMAP_DATA function.

### **Return values**

This function returns a hexadecimal string that can be used as a  $\verb"uuid"$  value.



 ${\sf REMAP\_DATA} \ in \ the \ \textit{Oracle} \& \ \textit{Database Utilities Guide}.$ 

