# 238
# DBMS_XMLSCHEMA_ANNOTATE

The DBMS_XMLSCHEMA_ANNOTATE package provides an interface to manage and configure the structured storage model, mainly through the use of pre-registration schema annotations.

This chapter contains the following topics:

- Overview
- Security Model
- Summary of DBMS_XMLSCHEMA_ANNOTATE Subprograms

> **See Also:**
>
> *Oracle XML DB Developer's Guide*

## DBMS_XMLSCHEMA_ANNOTATE Overview

The DBMS_XMLSCHEMA_ANNOTATE package contains procedures to manage and configure the structured storage model, mainly through the use of pre-registration schema annotations.

Schema annotations influence the way the XML data is stored. For example, the default table annotation assigns a user-provided name to an XML element instead of allowing the database to generate a system name. Consequently, query plans are more readable and it is easier to create constraints on that table.

## DBMS_XMLSCHEMA_ANNOTATE Security Model

Owned by XDB, the DBMS_XMLSCHEMA_ANNOTATE package must be created by SYS or XDB. The EXECUTE privilege is granted to PUBLIC. Subprograms in this package are executed using the privileges of the current user.

## Summary of DBMS_XMLSCHEMA_ANNOTATE Subprograms

This table lists and describes the DBMS_XMLSCHEMA_ANNOTATE package subprograms.

**Table 238-1 DBMS_XMLSCHEMA_ANNOTATE Package Subprograms**

| Subprogram | Description |
|---|---|
| ADDXDBNAMESPACE Procedure | Adds the XDB namespace required for XDB annotation |
| DISABLEDEFAULTTABLECREATION Procedure | Prevents the creation of a table for the top-level element by adding a default table attribute with an empty value to the element |
| DISABLEMAINTAINDOM Procedure | Sets the DOM fidelity attribute to FALSE |

**Table 238-1 (Cont.) DBMS_XMLSCHEMA_ANNOTATE Package Subprograms**

| Subprogram | Description |
|---|---|
| ENABLEDEFAULTTABLECREATION Procedure | Enables the creation of `ALL` top level tables by removing the empty default table name annotation |
| ENABLEMAINTAINDOM Procedure | Sets the DOM fidelity attribute to `TRUE` |
| GETSCHEMAANNOTATIONS Function | Creates a document containing the differences between the annotated XML schema and the original XML schema |
| GETSIDXDEFFROMVIEW Function | Takes a XMLTABLE view definition on a xmltype column or table and it returns a `CLOB` which can be used as parameter to create a structured xmlindex that backs up the XMLTABLE view as relational table |
| PRINTWARNINGS Procedure | Lets a user raise or suppress a warning if an annotation maps to zero nodes in the XML schema |
| REMOVEANYSTORAGE Procedure | Removes the setting of the SQL type from the `ANY` child of the complex type with the given name |
| REMOVEDEFAULTTABLE Procedure | Removes any default table attribute given for the element. After calling this procedure, the system generates table names |
| REMOVEMAINTAINDOM Procedure | Removes all annotations used to maintain DOM from the given schema |
| REMOVEOUTOFLINE Procedure | Removes any existing `SQLInline` attributes to prevent out-of-line storage |
| REMOVESQLCOLLTYPE Procedure | Removes a SQL collection type. |
| REMOVESQLNAME Procedure | Removes a `SQLNAME` from a global element |
| REMOVESQLTYPE Procedure | Removes a SQL type |
| REMOVESQLTYPEMAPPING Procedure | Removes the SQL type mapping for the given schema type. |
| REMOVETABLEPROPS Procedure | Removes the table storage properties from the `CREATE TABLE` statement |
| REMOVETIMESTAMPWITHTIMEZONE Procedure | Removes he setting of the `TimeStampWithTimeZone` datatype from all `dateTime` typed elements in the XML schema |
| SETANYSTORAGE Procedure | Assigns a SQL datatype to the `ANY` child of the complex type with the given name |
| SETDEFAULTTABLE Procedure | Sets the name of the table for the specified global element |
| SETOUTOFLINE Procedure | Sets the `SQLInline` attribute to `FALSE` |
| SETSCHEMAANNOTATATIONS Procedure | Takes the annotated differences resulting from a call to `DBMS_XMLSCHEMA_ANNOTATE.GETSCHEMAANNOTATIONS` and patches them into the provided XML schema |
| SETSQLCOLLTYPE Procedure | Assigns a SQL type name for a collection |
| SETSQLNAME Procedure | Assigns a name to the SQL attribute that corresponds to an element defined in the XML schema |
| SETSQLTYPE Procedure | Assigns a SQL type to a global object |
| SETSQLTYPEMAPPING Procedure | Defines a mapping of schema type and SQL type |
| SETTABLEPROPS Procedure | Specifies properties in the `TABLE` storage clause that is appended to the default `CREATE TABLE` statement |
| SETTIMESTAMPWITHTIMEZONE Procedure | Sets the `TIMESTAMPWITHTIMEZONE` datatype to all `dateTime` typed elements in the XML schema |

# ADDXDBNAMESPACE Procedure

This procedure adds the XDB namespace required for XDB annotation.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.ADDXDBNAMESPACE (
   xmlschema      IN OUT XMLTYPE);
```

### Parameters

**Table 238-2 ADDXDBNAMESPACE Procedure Parameters**

| Parameter | Description |
|---|---|
| xmschema | Gets an XML Schema as XMLTYPE, performs the annotation and returns it |

### Usage Notes

This procedure is called implicitly by any other procedure that adds a schema annotation. Since there is no reason to add an XDB namespace without other annotations, this procedure is most likely called by other annotations procedures and not by the user directly.

# DISABLEDEFAULTTABLECREATION Procedure

This procedure prevents the creation of a table for the top-level element by adding a default table attribute with an empty value to the element. The first overload applies to a specified top-level element and the second applies to all top-level elements. The procedure always overwrites. This is equivalent to using the schema annotation `xdb:defaultTable=""` for the top-level element or elements.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.DISABLEDEFAULTTABLECREATION (
   xmlschema         IN OUT XMLType,
   globalElementName IN VARCHAR2);


DBMS_XMLSCHEMA_ANNOTATE.DISABLEDEFAULTTABLECREATION (
   xmlschema         IN OUT XMLType);
```

### Parameters

**Table 238-3 DISABLEDEFAULTTABLECREATION Procedure Parameters**

| Parameter | Description |
|---|---|
| xmlschema | XML schema to be annotated |
| globalElementName | Name of the global element in the schema |

### Example

The `purchaseOrder` element will have an annotation similar to `xdb:defaultTable=""`.

```
DECLARE
   xml_schema   XMLTYPE;
BEGIN
   SELECT out INTO xml_schema FROM annotation_tab;
   DBMS_XMLSCHEMA_ANNOTATE.DISABLEDEFAULTTABLECREATION(xml_schema,
                                                       'purchaseOrder');
   UPDATE annotation_tab SET out = xml_schema;
END;
/
```

# DISABLEMAINTAINDOM Procedure

This procedure sets the DOM fidelity attribute to `FALSE`.

There are two overloads. The first sets DOM fidelity attribute to `FALSE` for all complex types, and the second sets it to `FALSE` for the named complex type. This is equivalent to adding `xdb:maintainDOM="false"` on all or specified complex types respectively.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.DISABLEMAINTAINDOM (
   xmlschema       IN OUT XMLType,
   overwrite       IN BOOLEAN default TRUE);


DBMS_XMLSCHEMA_ANNOTATE.DISABLEMAINTAINDOM (
   xmlschema       IN OUT XMLType,
   complexTypeName IN VARCHAR2,
   overwrite       IN BOOLEAN default TRUE);
```

### Parameters

**Table 238-4 DISABLEMAINTAINDOM Procedure Parameters**

| Parameter | Description |
|---|---|
| `xmlschema` | The XML schema to be annotated |
| `complexTypeName` | The name of the complex type |
| `overwrite` | A boolean that indicates whether or not the procedure overwrites element attributes. The default is `TRUE` |

# ENABLEDEFAULTTABLECREATION Procedure

This procedure enables the creation of ALL top level tables by removing the empty default table name annotation.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.ENABLEDEFAULTTABLECREATION (
   xmlschema            IN OUT  XMLTYPE);

DBMS_XMLSCHEMA_ANNOTATE.ENABLEDEFAULTTABLECREATION (
   xmlschema            IN OUT  XMLTYPE,
   globalElementName    IN      VARCHAR2););
```

### Parameters

**Table 238-5 ENABLEDEFAULTTABLECREATION Procedure Parameters**

| Parameter | Description |
|---|---|
| xmlschema | The XML schema to be annotated |
| gloablElementName | Name of the global element in the schema |

### Usage Notes

This procedure does not affect elements that have a default table name.

## ENABLEMAINTAINDOM Procedure

This overloaded procedure sets the DOM fidelity attribute to TRUE.

There are two overloads. The first sets DOM fidelity attribute to TRUE for all complex types, and the second sets it to TRUE for the named complex type.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.ENABLEMAINTAINDOM (
   xmlschema       IN OUT XMLType,
   overwrite       IN BOOLEAN default TRUE);


DBMS_XMLSCHEMA_ANNOTATE.ENABLEMAINTAINDOM (
   xmlschema        IN OUT XMLType,
   complexTypeName  IN VARCHAR2,
   overwrite        IN BOOLEAN default TRUE);
```

### Parameters

**Table 238-6 ENABLEMAINTAINDOM Procedure Parameters**

| Parameter | Description |
|---|---|
| xmlschema | The XML schema to be annotated |
| complexTypeName | The name of the complex type |
| overwrite | A boolean that indicates whether or not the procedure overwrites element attributes. The default is TRUE |

## GETSCHEMAANNOTATIONS Function

This function creates a document containing the differences between the annotated XML schema and the original XML schema.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.GETSCHEMAANNOTATIONS (
   xmlschema IN xmlType)
  RETURN XMLType;
```

### Parameters

**Table 238-7 GETSCHEMAANNOTATIONS Function Parameters**

| Parameter | Description |
|---|---|
| xmlschema | The original XML schema |

### Return Values

This function returns the document `annotations.xml` as an `XMLType`.

### Usage Notes

This function saves all annotations in one document, named `annotations`, and returns it. With this document, you can apply all annotations to a non-annotated schema, using `DBMS_XMLSCHEMA_ANNOTATE.GETSCHEMAANNOTATIONS`.

`DBMS_XMLSCHEMA_ANNOTATE.GETSCHEMAANNOTATIONS` is not available on Oracle Database release 10.2 (only Oracle Database release 11.*x*).

> **See Also:**
>
> SETSCHEMAANNOTATATIONS Procedure

### Example

For an example of `DBMS_XMLSCHEMA_ANNOTATE.GETSCHEMAANNOTATIONS`, see the example in SETSCHEMAANNOTATATIONS Procedure.

## GETSIDXDEFFROMVIEW Function

This function takes a XMLTABLE view definition on a xmltype column or table and it returns a `CLOB` which can be used as parameter to create a structured xmlindex that backs up the XMLTABLE view as relational table.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.GETSIDXDEFFROMVIEW (
   viewName   IN xmlType)
  RETURN CLOB;
```

### Parameters

**Table 238-8 GETSIDXDEFFROMVIEW Function Parameters**

| Parameter | Description |
|---|---|
| viewName | The original XML schema |

### Return Values

This function returns a `CLOB` which can be used as parameter to create a structured xmlindex that backs up the `XMLTABLE` view as relational table.

## PRINTWARNINGS Procedure

This procedure lets a user raise or suppress a warning if an annotation maps to zero nodes in the XML schema.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.PRINTWARNINGS (
   value       IN   BOOLEAN DEFAULT TRUE);
```

### Parameters

**Table 238-9 PRINTWARNINGS Procedure Parameters**

| Parameter | Description |
|---|---|
| `val` | For the `NO MATCHING ELEMENTS FOUND` error message to be raised `val` must be set to `TRUE`. In cases in which user wishes to suppress this warning, set to `FALSE`. |

### Usage Notes

If an annotation maps to more than one node in the XML schema, this raise the error `ANNOTATION MAPS TO MULTIPLE ELEMENTS`. In this case no annotation is performed, and the user must correct the parameters to the procedure call to refer to a unique node in the XML schema.

## REMOVEANYSTORAGE Procedure

This procedure removes the setting of the SQL type from the `ANY` child of the complex type with the given name.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.REMOVEANYSTORAGE (
   xmlschema       IN OUT XMLType,
   complexTypeName IN VARCHAR2);
```

### Parameters

**Table 238-10 REMOVEANYSTORAGE Procedure Parameters**

| Parameter | Description |
|---|---|
| `xmlschema` | The XML schema to be annotated. |
| `complexTypeName` | The name of the complex type. |

### Usage Notes

This procedure reverses the SETANYSTORAGE Procedure.

# REMOVEDEFAULTTABLE Procedure

This procedure removes any default table attribute given for the element.

After calling this procedure, the system generates table names. This procedure always overwrites.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.REMOVEDEFAULTTABLE (
   xmlschema          IN OUT XMLTYPE,
   globalElementName  IN     VARCHAR2);
```

### Parameters

**Table 238-11 REMOVEDEFAULTTABLE Procedure Parameters**

| Parameter | Description |
|---|---|
| xmlschema | XML schema to be annotated |
| globalElementName | Name of the global element in the schema |

### Example

Annotations can be verified anytime using "`select out from annotation_tab`".

```
--The purchaseOrder element will have no annotation for defaultTable.
DECLARE
   xml_schema XMLTYPE;
BEGIN
   SELECT out INTO xml_schema FROM annotation_tab;
   DBMS_XMLSCHEMA_ANNOTATE.REMOVEDEFAULTTABLE(xml_schema,
                                              'purchaseOrder');
   UPDATE annotation_tab SET out = xml_schema;
END;
/
```

# REMOVEMAINTAINDOM Procedure

This procedure removes all annotations used to maintain DOM from the given schema.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.REMOVEMAINTAINDOM (
   xmlschema    IN OUT XMLType);
```

### Parameters

**Table 238-12 REMOVEMAINTAINDOM Procedure Parameters**

| Parameter | Description |
|---|---|
| xmlschema | The XML schema to be annotated |

# REMOVEOUTOFLINE Procedure

This procedure removes any existing `SQLInline` attributes to prevent out-of-line storage.

There are three overloads.

### Syntax

Removes the `SQLInline` attribute for the named element.

```
DBMS_XMLSCHEMA_ANNOTATE.REMOVEOUTOFLINE (
   xmlschema           IN OUT  XMLType,
   elementName         IN      VARCHAR2,
   elementType         IN      VARCHAR2,
   overwrite           IN      BOOLEAN default TRUE);
```

Removes the `SQLInline` attribute for the object specified by its global object and local element names.

```
DBMS_XMLSCHEMA_ANNOTATE.REMOVEOUTOFLINE (
   xmlschema          IN OUT  XMLType,
   globalObject       IN      VARCHAR2,
   globalObjectName   IN      VARCHAR2,
   localElementName   IN      VARCHAR2);
```

Removes the `SQLInline` attribute for the referenced global element.

```
DBMS_XMLSCHEMA_ANNOTATE.REMOVEOUTOFLINE (
   xmlschema         IN OUT  XMLType,
   reference         IN      VARCHAR2);
```

### Parameters

**Table 238-13 REMOVEOUTOFLINE Procedure Parameters**

| Parameter | Description |
|---|---|
| `xmlschema` | The XML schema to be annotated |
| `elementName` | The element name |
| `elementType` | The element type |
| `globalObject` | The global object (global complex type or global element) |
| `globalObjectName` | The name of the global object |
| `localElementName` | The name of a local element that descends from the global element |
| `reference` | A reference to a global element |
| `overwrite` | A boolean that indicates whether or not the procedure overwrites element attributes. The default is `TRUE`. |

### Usage Notes

This procedure reverses SETOUTOFLINE Procedure.

# REMOVESQLCOLLTYPE Procedure

This procedure removes a SQL collection type.

The first overload removes the SQL collection type corresponding to the named element and the second overload removes the type from the XML element inside the complex type.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.REMOVESQLCOLLTYPE (
   xmlschema   IN OUT XMLType,
   elementName IN VARCHAR2);

DBMS_XMLSCHEMA_ANNOTATE.REMOVESQLCOLLTYPE (
   xmlschema        IN OUT XMLType,
   globalObject     IN VARCHAR2,
   globalName       IN VARCHAR2,
   localElementName IN VARCHAR2);
```

### Parameters

**Table 238-14 REMOVESQLCOLLTYPE Procedure Parameters**

| Parameter | Description |
|---|---|
| `xmlschema` | The XML schema to be annotated |
| `elementName` | The element name |
| `globalObject` | The global object (global complex type or global element) |
| `globalName` | The name of the global object |
| `localElementName` | The name of a local element that descends from the global element |

### Usage Notes

This procedure reverses the SETSQLCOLLTYPE Procedure.

# REMOVESQLNAME Procedure

This procedure removes a `SQLNAME` from a global element.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.REMOVESQLNAME (
   xmlschema          IN OUT  XMLType,
   globalObject       IN      VARCHAR2,
   globalObjectName   IN      VARCHAR2,
   localObject        IN      VARCHAR2,
   localObjectName    IN      VARCHAR2,
   sqlName            IN      VARCHAR2,
   overwrite          IN      BOOLEAN DEFAULT TRUE);
```

### Parameters

**Table 238-15 REMOVESQLNAME Procedure Parameters**

| Parameter | Description |
|---|---|
| xmlschema | XML schema to be annotated |
| globalObject | Global object (global complex type or global element) |
| globalObjectName | Name of the global object |
| localObject | Object descended from the global object |
| localObjectName | Name of the local object |
| sqlName | Name of the SQL attribute that corresponds to the element defined in the XML schema |
| overwrite | Boolean that indicates whether or not the procedure overwrites element attributes. The default is TRUE. |

### Example

The shipTo element will have an annotation similar to xdb:SQLName="SHIPTO_SQLNAME".

```
DECLARE
   xml_schema   XMLTYPE;
BEGIN
   SELECT out INTO xml_schema FROM annotation_tab;
   DBMS_XMLSCHEMA_ANNOTATE.SETSQLNAME (xml_schema,
                                       'element', 'purchaseOrder',
                                       'element', 'shipTo',
                                       'SHIPTO_SQLNAME');
   UPDATE annotation_tab SET out = xml_schema;
END;
/
```

## REMOVESQLTYPE Procedure

This procedure removes a SQL type.

The first overload removes a SQL type from a global element and the second overload removes the type from a global element inside the complex type.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.REMOVESQLTYPE (
   xmlschema in out XMLType,
   globalElementName   IN      VARCHAR2);

DBMS_XMLSCHEMA_ANNOTATE.REMOVESQLTYPE (
   xmlschema           IN OUT  XMLTYPE,
   globalObject        IN      VARCHAR2,
   globalObjectName    IN      VARCHAR2,
   localObject         IN      VARCHAR2,
   localObjectName     IN      VARCHAR2);
```

### Parameters

**Table 238-16 REMOVESQLTYPE Procedure Parameters**

| Parameter | Description |
|---|---|
| xmlschema | XML schema to be annotated. |
| globalObject | Global object (global complex type or global element) |
| globalElementName | Name of the global element. |
| globalObjectName | Name of the global object |
| localObject | Object descended from the global object |
| localObjectName | Name of the local object |

### Usage Notes

This procedure reverses the SETSQLTYPE Procedure.

## REMOVESQLTYPEMAPPING Procedure

This procedure removes the SQL type mapping for the given schema type.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.REMOVESQLTYPEMAPPING (
   xmlschema         IN OUT  XMLTYPE,
   schemaTypeName    IN      VARCHAR2);
```

### Parameters

**Table 238-17 REMOVESQLTYPEMAPPING Procedure Parameters**

| Parameter | Description |
|---|---|
| xmlschema | XML schema to be annotated |
| schemaTypeName | Name of the schema type |

### Usage Notes

This procedure reverses the SETSQLTYPEMAPPING Procedure.

## REMOVETABLEPROPS Procedure

This procedure removes the table storage properties from the `CREATE TABLE` statement.

This procedure is overloaded. Each overload has different parameter requirements as indicated.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.REMOVETABLEPROPS (
   xmlschema            IN OUT  XMLTYPE,
   globalElementName    IN      VARCHAR2);
```

```
DBMS_XMLSCHEMA_ANNOTATE.REMOVETABLEPROPS (
   xmlschema            IN OUT  XMLTYPE,
   globalObject         IN      VARCHAR2,
   globalObjectName     IN      VARCHAR2,
   localElementName     IN      VARCHAR2);
```

### Parameters

**Table 238-18 REMOVETABLEPROPS Procedure Parameters**

| Parameter | Description |
|---|---|
| `xmlschema` | XML schema to be annotated |
| `globalElementName` | Name of the global element in the schema |
| `globalObject` | Global object (global complex type or global element) |
| `globalObjectName` | Name of the global object |
| `localElementName` | Name of a local element that descends from the global element |

### Usage Notes

This procedure reverses the SETTABLEPROPS Procedure.

## REMOVETIMESTAMPWITHTIMEZONE Procedure

This procedure removes the setting of the `TimeStampWithTimeZone` datatype from all `dateTime` typed elements in the XML schema.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.REMOVETIMESTAMPWITHTIMEZONE (
   xmlschema          IN OUT  XMLTYPE);

DBMS_XMLSCHEMA_ANNOTATE.REMOVETIMESTAMPWITHTIMEZONE (
   xmlschema         IN OUT  XMLTYPE,
   schemaTypeName    IN      VARCHAR2);
```

### Parameters

**Table 238-19 REMOVETIMESTAMPWITHTIMEZONE Procedure Parameters**

| Parameter | Description |
|---|---|
| `xmlschema` | XML schema to be annotated |
| `schemaTypeName` | Name of the schema type |

### Usage Notes

This procedure reverses the SETTIMESTAMPWITHTIMEZONE Procedure.

# SETANYSTORAGE Procedure

This procedure assigns a SQL datatype to the `ANY` child of the complex type with the given name.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.SETANYSTORAGE )
   xmlschema       IN OUT  XMLType,
   complexTypeName IN      VARCHAR2,
   sqlTypeName     IN      VARCHAR2,
   overwrite       IN      BOOLEAN DEFAULT TRUE);
```

### Parameters

**Table 238-20 SETANYSTORAGE Procedure Parameters**

| Parameter | Description |
|---|---|
| `xmlschema` | XML schema to be annotated |
| `complexTypeName` | Name of the complex type |
| `sqlTypeName` | Name of the SQL type |
| `overwrite` | Boolean that indicates whether or not the procedure overwrites element attributes. The default is `TRUE`. |

### Example

The `xsd:any child` of complex type `Items` is assigned an annotation similar to `xdb:SQLType="VARCHAR"`.

```
DECLARE   xml_schema  XMLTYPE;BEGIN   SELECT out INTO xml_schema FROM annotation_tab;
DBMS_XMLSCHEMA_ANNOTATE.setAnyStorage
(xml_schema,
'Items',                                          'VARCHAR');   UPDATE annotation_tab SET
out = xml_schema;END;
/
```

# SETDEFAULTTABLE Procedure

This procedure sets the name of the table for the specified global element. This is equivalent to using the schema annotation `xdb:defaultTable="<default_table_name>"` for the top-level element.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.SETDEFAULTTABLE (
   xmlschema           IN  OUT  XMLTYPE,
   globalElementName   IN       VARCHAR2,
   tableName           IN       VARCHAR2,
   overwrite           IN       BOOLEAN   DEFAULT TRUE);
```

### Parameters

**Table 238-21 SETDEFAULTTABLE Procedure Parameters**

| Parameter | Description |
|---|---|
| xmlschema | XML schema to be annotated |
| globalElementName | Name of the global element in the schema |
| tableName | Name being assigned to the table |
| overwrite | Boolean that indicates whether or not the procedure overwrites element attributes. The default is TRUE. |

## SETOUTOFLINE Procedure

This procedure sets the SQLInline attribute to FALSE, that is, it sets xdb:SQLInLine=FALSE.

This forces XDB to store the corresponding elements in the XML document out-of-line as rows in a separate XMLType table. XDB stores references to each row of the XMLType table in a link table that is maintained by the main table

This procedure can improve performance in some situations if the out-of-line table acts as the driver for the query. Storing elements in an out-of-line table also reduces the numbers of columns in the base table, thus avoiding '4096 column limit' errors during XML schema registration, when some elements have complex types with many elements.

> **Also See:**
>
> *Oracle XML DB Developer's Guide*

There are three overloads.

### Syntax

Sets the SQLInline attribute to FALSE, forcing out-of-line storage for the named element.

```
DBMS_XMLSCHEMA_ANNOTATE.SETOUTOFLINE (
   xmlschema          IN OUT  XMLType,
   elementName        IN      VARCHAR2,
   elementType        IN      VARCHAR2,
   defaultTableName   IN      VARCHAR2,
   overwrite          IN      BOOLEAN DEFAULT TRUE);
```

Sets the SQLInline attribute to FALSE, forcing out-of-line storage for the element specified by its local and global name.

```
DBMS_XMLSCHEMA_ANNOTATE.SETOUTOFLINE (
   xmlschema          IN OUT  XMLType,
   globalObject       IN      VARCHAR2,
   globalObjectName   IN      VARCHAR2,
   localElementName   IN      VARCHAR2,
   defaultTableName   IN      VARCHAR2,
   overwrite          IN      BOOLEAN DEFAULT TRUE);
```

Sets the `SQLInline` attribute to `FALSE` to force out-of-line storage and sets the default table name for all references to a particular global element.

```
DBMS_XMLSCHEMA_ANNOTATE.SETOUTOFLINE (
   xmlschema         IN OUT  XMLType,
   reference         IN      VARCHAR2,
   defaultTableName  IN      VARCHAR2,
   overwrite         IN      BOOLEAN DEFAULT TRUE);
```

### Parameters

**Table 238-22 SETOUTOFLINE Procedure Parameters**

| Parameter | Description |
|---|---|
| `xmlschema` | The XML schema to be annotated. |
| `elementName` | The element name |
| `elementType` | The element type |
| `defaultTableName` | The name of the default table. |
| `globalObject` | The global object (global complex type or global element) |
| `globalObjectName` | The name of the global object |
| `localElementName` | The name of a local element that descends from the global element. |
| `reference` | A reference to a global element |
| `overwrite` | A boolean that indicates whether or not the procedure overwrites element attributes. The default is `TRUE`. |

### Usage Notes

After XML schema registration and before loading XML instance data, use `DBMS_XMLSTORAGE_MANAGE.SCOPEXMLREFERENCES()` to make these references scope to the out-of-line table only. This ensures better query performance later on.

### Example

The following example illustrates the third overloaded method. The element comment will have an annotation similar to `xdb:defaultTable="CMMNT_DEFAULT_TABLE"`

```
DECLARE
   xml_schema xmltype;
BEGIN
SELECT OUT INTO xml_schema FROM annotation_tab;

DBMS_XMLSCHEMA_ANNOTATE.SETOUTOFLINE (xml_schema,
                                      'ipo:comment',
                                      'CMMNT_DEFAULT_TABLE');
UPDATE annotation_tab SET OUT = xml_schema;
END;
/
```

# SETSCHEMAANNOTATATIONS Procedure

This procedure takes the annotated differences resulting from a call to `DBMS_XMLSCHEMA_ANNOTATE.GETSCHEMAANNOTATIONS` and patches them into the provided XML schema.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.SETSCHEMAANNOTATIONS (
   xmlschema     IN OUT xmlType,
   annotations   IN VARCHAR2);
```

### Parameters

**Table 238-23 SETSCHEMAANNOTATIONS Procedure Parameters**

| Parameter | Description |
|---|---|
| `xmlschema` | An XML schema to be patched. |
| `annotations` | The differences document produced by calling `DBMS_XMLSCHEMA_ANNOTATE.GETSCHEMAANNOTATIONS` on the original XML schema and an annotated XML schema. |

### Usage Notes

`DBMS_XMLSCHEMA_ANNOTATE.SETSCHEMAANNOTATATIONS` is not available on Oracle Database release 10.2 (only Oracle Database release 11.*x*).

> **See Also:**
>
> GETSCHEMAANNOTATIONS Function

### Example

The following example illustrates `DBMS_XMLSCHEMA_ANNOTATE.SETSCHEMAANNOTATIONS` shown here and GETSCHEMAANNOTATIONS Function.

```
-- test getannotations and apply them
declare
 xml_schema xmltype;
 xml_schema2 xmltype;
 annotations xmltype;
begin
   select out into xml_schema from annotation_tab;

   -- get the annotations from the schema
   annotations := DBMS_XMLSCHEMA_ANNOTATE.getSchemaAnnotations (xml_schema);

   -- apply the annotations to the schema
   select inp into xml_schema2 from annotation_tab;

   DBMS_XMLSCHEMA_ANNOTATE.setSchemaAnnotations(xml_schema2, annotations);

   update annotation_tab t set t.out = xml_schema2;
```

```
end;
/
```

# SETSQLCOLLTYPE Procedure

This procedure assigns a SQL type name for a collection. A collection is a global or local element with `maxOccurs>1`.

Using this procedure, XDB creates `SQLType`s with the user-defined names provided.

There are two overloads. The first sets the name of the SQL collection type corresponding to an XML element and the second to an XML element inside the specified complex type.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.SETSQLCOLLTYPE (
   xmlschema          IN OUT   XMLTYPE,
   elementName        IN       VARCHAR2,
   sqlCollType        IN       VARCHAR2,
   overwrite          IN       BOOLEAN DEFAULT TRUE);

DBMS_XMLSCHEMA_ANNOTATE.SETSQLCOLLTYPE (
   xmlschema          IN OUT XMLType,
   globalObject       IN VARCHAR2,
   globalObjectName   IN VARCHAR2,
   localElementName   IN VARCHAR2,
   sqlCollType        IN VARCHAR2,
   overwrite          IN BOOLEAN default TRUE );
```

### Parameters

**Table 238-24 SETSQLCOLLTYPE Procedure Parameters**

| Parameter | Description |
|---|---|
| `xmlschema` | The XML schema to be annotated |
| `elementName` | The element name |
| `sqlCollType` | The SQL collection type |
| `globalObject` | The global object (global complex type or global element) |
| `globalObjectName` | The name of the global object |
| `localElementName` | The name of a local element that descends from the global element |
| `overwrite` | A boolean that indicates whether or not the procedure overwrites element attributes. The default is `TRUE`. |

### Example

The `item` element will have an annotation similar to `xdb:SQLCollType="ITEM_SQL_COL_TYPE"`.

```
declare
   xml_schema xmltype;
begin
   SELECT out INTO xml_schema FROM annotation_tab;
   DBMS_XMLSCHEMA_ANNOTATE.setSQLCollType (xml_schema,
                                           'item',
                                           'ITEM_SQL_COL_TYPE',TRUE);
   UPDATE annotation_tab SET out = xml_schema;
end;
```

# SETSQLNAME Procedure

This procedure assigns a name to the SQL attribute that corresponds to an element defined in the XML schema.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.SETSQLNAME (
   xmlschema          IN OUT  XMLType,
   globalObject       IN      VARCHAR2,
   globalObjectName   IN      VARCHAR2,
   localObject        IN      VARCHAR2,
   localObjectName    IN      VARCHAR2,
   sqlName            IN      VARCHAR2,
   overwrite          IN      BOOLEAN DEFAULT TRUE);
```

### Parameters

**Table 238-25 SETSQLNAME Procedure Parameters**

| Parameter | Description |
|---|---|
| `xmlschema` | XML schema to be annotated |
| `globalObject` | Global object (global complex type or global element) |
| `globalObjectName` | Name of the global object |
| `localObject` | Object descended from the global object |
| `localObjectName` | Name of the local object |
| `sqlName` | Name of the SQL attribute that corresponds to the element defined in the XML schema |
| `overwrite` | Boolean that indicates whether or not the procedure overwrites element attributes. The default is `TRUE`. |

### Example

The `shipTo` element will have an annotation similar to `xdb:SQLName="SHIPTO_SQLNAME"`.

```
DECLARE
   xml_schema   XMLTYPE;
BEGIN
   SELECT out INTO xml_schema FROM annotation_tab;
   DBMS_XMLSCHEMA_ANNOTATE.SETSQLNAME (xml_schema,
                                       'element', 'purchaseOrder',
                                       'element', 'shipTo',
                                       'SHIPTO_SQLNAME');
   UPDATE annotation_tab SET out = xml_schema;
END;
/
```

# SETSQLTYPE Procedure

This procedure assigns a SQL type to a global object.

There are two overloads. The first overload assigns a SQL Type to a global object, such as a global element or global complex type and the second to a local object.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.SETSQLTYPE (
   xmlschema            IN OUT   XMLTYPE,
   globalElementName    IN       VARCHAR2,
   sqlType              IN       VARCHAR2,
   overwrite            IN       BOOLEAN DEFAULT TRUE);


DBMS_XMLSCHEMA_ANNOTATE.SETSQLTYPE (
   xmlschema           IN OUT   XMLTYPE,
   globalObject        IN       VARCHAR2,
   globalObjectName    IN       VARCHAR2,
   localObject         IN       VARCHAR2,
   localObjectName     IN       VARCHAR2,
   sqlType             IN       VARCHAR2,
   overwrite           IN       BOOLEAN DEFAULT TRUE);
```

### Parameters

**Table 238-26 SETSQLTYPE Procedure Parameters**

| Parameter | Description |
|---|---|
| xmlschema | XML schema to be annotated |
| globalObject | Global object (global complex type or global element) |
| globalObjectName | Name of the global object |
| globalElementName | Name of the global element |
| localObject | Object descended from the global object |
| localObjectName | Name of the local object |
| sqlType | SQL type assigned to the named global element |
| overwrite | Boolean that indicates whether or not the procedure overwrites element attributes. The default is TRUE. |

### Example

The purchaseOrder element will have an annotation similar to xdb:SQLType="PO_SQLTYPE" and the shipTo element has one similar to xdb:SQLType="VARCHAR".

```
DECLARE
   xml_schema  xmltype;
BEGIN
  SELECT out INTO xml_schema FROM annotation_tab;
  DBMS_XMLSCHEMA_ANNOTATE.setSQLType (xml_schema,
                                      'purchaseOrder',
                                      'PO_SQLTYPE');
  UPDATE annotation_tab SET out = xml_schema;
END;
/

DECLARE   xml_schema   xmltype;BEGIN  SELECT out INTO xml_schema FROM annotation_tab;
DBMS_XMLSCHEMA_ANNOTATE.setSQLType (xml_schema,
'element','purchaseOrder',
'element' ,'shipTo',                                        'VARCHAR');  UPDATE
annotation_tab SET out = xml_schema;END;
/
```

# SETSQLTYPEMAPPING Procedure

This procedure defines a mapping of schema type and SQL type.

If you use this procedure, you do not need to call the `SETSQLTYPE` procedure on all instances of the schema type; instead the procedure traverses the schema and assigns the SQL type automatically.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.SETSQLTYPEMAPPING (
   xmlschema          IN OUT   XMLType,
   schemaTypeName     IN       VARCHAR2,
   sqlTypeName        IN       VARCHAR2,
   overwrite          IN       BOOLEAN DEFAULT TRUE);
```

### Parameters

**Table 238-27 SETSQLTYPEMAPPING Procedure Parameters**

| Parameter | Description |
|---|---|
| `xmlschema` | XML schema to be annotated |
| `schemaTypeName` | Schema type |
| `sqlTypeName` | Name of the SQL type |
| `overwrite` | Boolean that indicates whether or not the procedure overwrites element attributes. The default is `TRUE` |

### Example

The attribute `orderDate` will have an annotation similar to `xdb:SQLType="DATE"`.

```
declare   xml_schema xmltype;beginSELECT out INTO xml_schema FROM
annotation_tab;DBMS_XMLSCHEMA_ANNOTATE.setSQLTypeMapping
(xml_schema,
'date',                                          'DATE');UPDATE annotation_tab SET out =
xml_schema;end;
/
```

# SETTABLEPROPS Procedure

This procedure specifies properties in the `TABLE` storage clause that is appended to the default `CREATE TABLE` statement.

There are two overloads with different parameter requirements, as indicated:

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.SETTABLEPROPS (
   xmlschema            IN OUT  XMLType,
   globalElementName    IN      VARCHAR2,
   tableProps           IN      VARCHAR2,
   overwrite            IN      BOOLEAN DEFAULT TRUE);


DBMS_XMLSCHEMA_ANNOTATE.SETTABLEPROPS (
   xmlschema            IN OUT  XMLTYPE,
```

```
   globalObject          IN      VARCHAR2,
   globalObjectName      IN      VARCHAR2,
   localElementName      IN      VARCHAR2,
   tableProps            IN      VARCHAR2,
   overwrite             IN      BOOLEAN DEFAULT TRUE);
```

### Parameters

**Table 238-28 SETTABLEPROPS Procedure Parameters**

| Parameter | Description |
|---|---|
| xmlschema | XML schema to be annotated |
| globalElementName | Name of the global element in the schema |
| tableProps | Table properties |
| globalObject | Global object (global complex type or global element) |
| globalObjectName | Name of the global object |
| localElementName | Name of a local element that descends from the global element |
| overwrite | Boolean that indicates whether or not the procedure overwrites element attributes. The default is TRUE. |

### Example

The purchaseOrder element will have an annotation similar to xdb:tableProps="CACHE".

```
DECLARE   xml_schema XMLTYPE;BEGIN   SELECT out INTO xml_schema FROM annotation_tab;
DBMS_XMLSCHEMA_ANNOTATE.SETTABLEPROPS(xml_schema,
 'purchaseOrder' ,'CACHE');   UPDATE annotation_tab SET out = xml_schema;END;
/
```

## SETTIMESTAMPWITHTIMEZONE Procedure

This procedure sets the TIMESTAMPWITHTIMEZONE datatype to all dateTime typed elements in the XML schema.

This is equivalent to adding xdb:SQLType="TIMESTAMP WITH TIME ZONE" to all dateTime objects.

### Syntax

```
DBMS_XMLSCHEMA_ANNOTATE.SETTIMESTAMPWITHTIMEZONE (
   xmlschema      IN OUT   XMLTYPE,
   overwrite      IN       BOOLEAN DEFAULT TRUE);
```

### Parameters

**Table 238-29 SETTIMESTAMPWITHTIMEZONE Procedure Parameters**

| Parameter | Description |
|---|---|
| xmlschema | XML schema to be annotated |
| overwrite | Boolean that indicates whether or not the procedure overwrites element attributes. The default is TRUE. |