


# DBMS\_XMLINDEX

The `DBMS_XMLINDEX` package provides an interface to implement asynchronous indexing.

This chapter contains the following topics:

- [Overview](#)
- [Security Model](#)
- [Summary of DBMS\\_XMLINDEX Subprograms](#)

 **See Also:**

*Oracle XML DB Developer's Guide* for more information about "XMLIndex"

## DBMS\_XMLINDEX Overview

`DBMS_XMLINDEX` provides a mechanism for asynchronous index maintenance.

### Asynchronous Index Maintenance

The basic XMLIndex is maintained on every DML operation. However, given the computing costs, in many cases the availability of stale result is adequate. In such situations, it is desirable to defer index updates to a more convenient time, for example when the load on the database is low. `DBMS_XMLINDEX` provides this mechanism.

## DBMS\_XMLINDEX Security Model

Owned by `XDB`, the `DBMS_XMLINDEX` 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\_XMLINDEX Subprograms

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

**Table 235-1 DBMS\_XMLINDEX Package Subprograms**

Subprogram	Description
<a href="#">CREATEDATEINDEX Procedure</a>	Creates a secondary index for date values in the <code>VALUE</code> column of a <code>PATH TABLE</code> which is the storage table of an XMLIndex
<a href="#">CREATENUMBERINDEX Procedure</a>	Creates a secondary index for number values in the <code>VALUE</code> column of a <code>PATH TABLE</code> which is the storage table of an XMLIndex

**Table 235-1 (Cont.) DBMS\_XMLINDEX Package Subprograms**

Subprogram	Description
<a href="#">DROPPARAMETER Procedure</a>	Drops the XMLIndex parameter string that is associated with a given parameter identifier.
<a href="#">MODIFYPARAMETER Procedure</a>	Modifies the XMLIndex parameter string that is associated with a given parameter name
<a href="#">PROCESS_PENDING Procedure</a>	Processes pending rows for a NONBLOCKING ALTER INDEX OPERATION on an XMLIndex
<a href="#">REGISTERPARAMETER Procedure</a>	Registers a parameter string and XMLIndex parameter string pair in XDB
<a href="#">SYNCINDEX Procedure</a>	Synchronizes the index manually

## CREATEDATEINDEX Procedure

This procedure creates a secondary index for date values in the `VALUE` column of a `PATH TABLE` which is the storage table of an XMLIndex. The second form of the procedure allows for the `date_index_clause` to be set to an empty string.

### Syntax

```
DBMS_XMLINDEX.CREATEDATEINDEX (
    xml_index_schema IN VARCHAR2,
    xml_index_name   IN VARCHAR2,
    date_index_name  IN VARCHAR2,
    xmltypename      IN VARCHAR2,
    date_index_clause IN VARCHAR2);
```

```
DBMS_XMLINDEX.CREATEDATEINDEX (
    xml_index_schema IN VARCHAR2,
    xml_index_name   IN VARCHAR2,
    date_index_name  IN VARCHAR2,
    xmltypename      IN VARCHAR2);
```

### Parameters

**Table 235-2 CREATEDATEINDEX Procedure Parameters**

Parameter	Description
<code>xml_index_schema</code>	Name of the owner of the XMLIndex
<code>xml_index_name</code>	Name of the XMLIndex
<code>date_index_name</code>	Name of the secondary index to be created for date values in the <code>VALUE</code> column of the <code>PATH TABLE</code> of XMLIndex named <code>xml_index_name</code> and owned by <code>xml_index_schema</code>
<code>xmltypename</code>	The type to which values in the <code>VALUE</code> column of the path table are to be cast. Acceptable values are the following strings: <code>DATETIME</code> , <code>TIME</code> , <code>DATE</code> , <code>GDAY</code> , <code>GMONTH</code> , <code>GYEAR</code> , <code>GYEARMONTH</code> , <code>GMONTHDAY</code> .
<code>date_index_clause</code>	Storage clause to be applied to the date index during its creation. This is a string argument appended to the <code>CREATE INDEX</code> statement for creating the date index

## CREATENUMBERINDEX Procedure

This procedure creates a secondary index for number values in the `VALUE` column of a `PATH TABLE` which is the storage table of an `XMLIndex`.

### Syntax

```
DBMS_XMLINDEX.CREATENUMBERINDEX (  
    xml_index_schema    IN    VARCHAR2,  
    xml_index_name      IN    VARCHAR2,  
    num_index_name      IN    VARCHAR2,  
    num_index_clause    IN    VARCHAR2,  
    xmltypename         IN    VARCHAR2);
```

### Parameters

**Table 235-3 CREATENUMBERINDEX Procedure Parameters**

Parameter	Description
<code>xml_index_schema</code>	Name of the owner of the <code>XMLIndex</code>
<code>xml_index_name</code>	Name of the <code>XMLIndex</code>
<code>num_index_name</code>	Name of the secondary index to be created for number values in the <code>VALUE</code> column of the <code>PATH TABLE</code> of <code>XMLIndex</code> named <code>xml_index_name</code> and owned by <code>xml_index_schema</code>
<code>num_index_clause</code>	Storage clause to be applied to the number index during its creation. This is a string argument appended to the <code>CREATE INDEX</code> statement for creating the number index.
<code>xmltypename</code>	The type to which values in the <code>VALUE</code> column of the path table are to be cast. Acceptable values are the following strings: <code>FLOAT</code> , <code>DOUBLE</code> , <code>DECIMAL</code> , <code>INTEGER</code> , <code>NONPOSITIVEINTEGER</code> , <code>NEGATIVEINTEGER</code> , <code>LONG</code> , <code>INT</code> , <code>SHORT</code> , <code>BYTE</code> , <code>NONNEGATIVEINTEGER</code> , <code>UNSIGNEDLONG</code> , <code>UNSIGNEDINT</code> , <code>UNSIGNEDSHORT</code> , <code>UNSIGNEDBYTE</code> , <code>POSITIVEINTEGER</code> .

## DROPPARAMETER Procedure

This procedure drops the `XMLIndex` parameter string that is associated with a given parameter identifier.

### Syntax

```
DBMS_XMLINDEX.DROPPARAMETER (  
    name    IN    VARCHAR2);
```

### Parameters

**Table 235-4 DROPPARAMETER Procedure Parameters**

Parameter	Description
<code>name</code>	Identifier for parameter string

## Examples

```
DBMS_XMLINDEX.DROPPARAMETER (  
    'myIndexParam');
```

## MODIFYPARAMETER Procedure

This procedure modifies the `XMLIndex` parameter string that is associated with a given parameter identifier.

### Syntax

```
DBMS_XMLINDEX.MODIFYPARAMETER (  
    name          IN          VARCHAR2,  
    parameter     IN          CLOB);
```

### Parameters

**Table 235-5** MODIFYPARAMETER Procedure Parameters

Parameter	Description
name	Identifier for parameter string
parameter	XMLIndex parameter clause that can appear in a CREATE INDEX or an ALTER INDEX statement

### Examples

```
DBMS_XMLINDEX.MODIFYPARAMETER (  
    'myIndexParam',  
    'PATH TABLE po_ptab  
    PATH ID INDEX po_pidx  
    ORDER KEY INDEX po_oidx  
    VALUE INDEX po_vidx');
```

## PROCESS\_PENDING Procedure

This procedure processes executes DMLs required to complete a NONBLOCKING ALTER INDEX ADD\_GROUP/ADD\_COLUMN operation on an XMLIndex.

### Syntax

```
DBMS_XMLINDEX.PROCESS_PENDING (  
    xml_index_schema IN          VARCHAR2,  
    xml_index_name   IN          VARCHAR2,  
    pending_row_count OUT        BINARY_INTEGER,  
    error_row_count   OUT        BINARY_INTEGER);
```

### Parameters

**Table 235-6** PROCESS\_PENDING Procedure Parameters

Parameter	Description
xml_index_schema	Name of the owner of the XMLIndex

**Table 235-6 (Cont.) PROCESS\_PENDING Procedure Parameters**

Parameter	Description
xml_index_name	Name of the XMLIndex to be altered using NONBLOCKING ALTER INDEX OPERATION
pending_row_count	Number of pending rows to be processed
error_row_count	Number of rows for which indexing may have failed because of an error

#### Usage Notes

- This procedure will iteratively attempt to index all necessary rows in small batches while skipping rows that are locked and rows for which index maintenance fails with an error. Therefore, it may have to be executed multiple times for an XMLIndex until all pending rows are processed. Once all pending rows are processed, user can complete the NONBLOCKING ALTER INDEX OPERATION.
- If it is not possible process all the pending rows after multiple trials, the user will have to manually triage the locking or error issues by examining unprocessed rows in SYS\_AIXSXI\_#####\_PENDINGTAB and errors in SYS\_AIXSXI\_#####\_ERRORTAB.. Keeping track of rows and the errors is useful in triaging issues.

#### Examples

```
EXEC DBMS_XMLINDEX.PROCESS_PENDING(  
    'SCOTT', 'PO_XMLINDEX_IX', out_param1, out_param2);
```

## REGISTERPARAMETER Procedure

This procedure registers a parameter identifier and XMLIndex parameter string pair in XDB.

#### Syntax

```
DBMS_XMLINDEX.REGISTERPARAMETER (  
    name          IN      VARCHAR2,  
    parameter     IN      CLOB);
```

#### Parameters

**Table 235-7 REGISTERPARAMETER Procedure Parameters**

Parameter	Description
name	Identifier for parameter string
parameter	XMLIndex parameter clause that can appear in a CREATE INDEX or an ALTER INDEX statement

#### Examples

```
DBMS_XMLINDEX.REGISTERPARAMETER (  
    'myIndexParam',  
    'PATH TABLE po_ptab  
    PATH ID INDEX po_pidx  
    ORDER KEY INDEX po_oidx  
    VALUE INDEX po_vidx
```

```
PATHS (NAMESPACE MAPPING(xmlns:p="http://www.example.com/IPO"))
GROUP MASTERGROUP XMLTABLE PO_TAB
(''/p:PurchaseOrder''
  COLUMNS
    REFERENCE VARCHAR2(30) PATH ''p:Reference'',
    REQUESTOR VARCHAR2(30) PATH ''p:Requestor'' )
GROUP ITEMGROUP XMLTABLE ITEMGROUP_TAB
(''/p:PurchaseOrder/p:LineItems/p:LineItem''
  COLUMNS
    LINENUMBER NUMBER(38) PATH ''@p:ItemNumber'',
    QUANTITY NUMBER(38) PATH ''@p:Quantity'',
    DESCRIPTION VARCHAR2(256) PATH ''p:Description''));
```

## SYNCINDEX Procedure

This function synchronizes an asynchronously maintained XMLIndex.

It applies to the XMLIndex changes that are logged in the pending table, and brings the path table up-to-date with the base XMLTYPE column.

### Syntax

```
DBMS_XMLINDEX.SYNCINDEX (
  xml_index_schema      IN VARCHAR2,
  xml_index_name        IN VARCHAR2,
  partition_name        IN VARCHAR2 DEFAULT NULL,
  reindex               IN BOOLEAN DEFAULT FALSE);
```

### Parameters

**Table 235-8** SYNCINDEX Procedure Parameters

Parameter	Description
xml_index_schema	Name of the owner of the XMLIndex
xml_schema_name	Name of the XMLIndex
partition_name	[Currently not supported]
reindex	Default is FALSE. If set to TRUE, this drops the secondary indexes and recreates them later so that they can be bulk-loaded.

### Examples

```
EXEC DBMS_XMLINDEX.SYNCINDEX('USER1', 'SS_TAB_XMLI', REINDEX=>TRUE);
```