228
DBMS_XDB_REPOS

The DBMS_XDB_REPOS package provides an interface to operate on the Oracle XML database Repository.

This chapter contains the following topics:

- Overview
- Security Model
- Constants
- Summary of DBMS_XDB_REPOS Subprograms

> **See Also:**
>
> Oracle XML DB Developer's Guide for more information regarding:
>
> - Using and managing repository resources
> - ACL-based security management (controlling access to repository resources)
> - Managing XLink and XInclude links
> - Loading documents into the repository
> - Creating, deleting, and managing resource metadata

DBMS_XDB_REPOS Overview

The DBMS_XDB_REPOS package lets you operate on the Oracle XML DB Repository to create, modify and delete resources, including managing security based on access control lists (ACLs). The interface provides both query and DML functions.

Using a combination of PL/SQL packages - DBMS_XDB_REPOS, DBMS_XDBZ, and DBMS_XDB_VERSION - you can create, delete, and rename documents and folders, move a file or folder within the folder hierarchy, set and change the access permissions on a file or folder, and initiate and manage versioning.

DBMS_XDB_REPOS Security Model

Owned by XDB, the DBMS_XDB_REPOS 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. Subprograms that operate on the XDB Configuration will succeed only if the current user is SYS or XDB, or the current user has the XDBADMIN or DBA role.

DBMS_XDB_REPOS Constants

The DBMS_XDB_REPOS package defines several constants that can be used for specifying parameter values.

These constants are shown in the following table.

Table 228-1 DBMS_XDB_REPOS Constants

Constant	Type	Value	Description
DELETE_RESOURCE	NUMBER	1	Deletes a resource; fails if the resource has children.
DELETE_RECURSIVE	NUMBER	2	Deletes a resource and its children, if any.
DELETE_FORCE	NUMBER	3	Deletes the resource, even if the object it contains is invalid
DELETE_RECURSIVE_FORCE	NUMBER	4	Deletes a resource and its children, if any, even if the object it contains is invalid
DELETE_RES_METADATA_CASCADE	NUMBER	1	Deletes the row in the metadata
DELETE_RES_METADATA_NOCASCADE	NUMBER	2	Does not delete the row
DEFAULT_LOCK_TIMEOUT	PLS_INTEGER	(60*60)	Timeout value (in seconds) of the webdav lock
LINK_TYPE_HARD	NUMBER	1	Hard link of a folder to a resource
LINK_TYPE_WEAK	NUMBER	2	Weak link of a folder to a resource
LINK_TYPE_SYMBOLIC	NUMBER	3	Symbolic link of a folder to a resource

Summary of DBMS_XDB_REPOS Subprograms

This table lists the DBMS_XDB_REPOS subprograms and briefly describes them.

Table 228-2 DBMS_XDB_REPOS Package Subprograms

Subprogram	Description
ACLCHECKPRIVILEGES Function	Checks access privileges granted to the current user by specified ACL document on a resource whose owner is specified by the 'owner' parameter.
APPENDRESOURCEMETADATA Procedure	Takes in user-defined metadata either as a REF to XMLTYPE or an XMLTYPE and adds it to the desired resource
CHANGEOWNER Procedure	Changes the owner of the resource/s to the specified owner.
CHANGEPRIVILEGES Function	Adds a specified ACE to a specified resource's ACL
CHECKPRIVILEGES Function	Checks access privileges granted to the current user on the specified resource
CREATEFOLDER Function	Creates a new folder resource in the hierarchy
CREATEOIDPATH Function	Creates a virtual path to the resource based on object ID

Table 228-2 (Cont.) DBMS_XDB_REPOS Package Subprograms

Subprogram	Description
CREATERESOURCE Functions	Creates a new resource
DELETERESOURCE Procedure	Deletes a resource from the hierarchy
DELETERESOURCEMETADATA Procedures	Deletes metadata from a resource (can be used for schema-based or nonschema-based metadata)
EXISTSRESOURCE Function	Determines if a resource is the hierarchy, based on its absolute path
GETACLDOCUMENT Function	Retrieves ACL document that protects resource given its path name
GETCONTENTBLOB Function	Retrieves the contents of a resource returned as a BLOB
GETCONTENTCLOB Function	Retrieves the contents of a resource returned as a CLOB
GETCONTENTVARCHAR2 Function	Retrieves the contents of a resource returned as a string
GETCONTENTXMLREF Function	Retrieves the contents of a resource returned as a a `REF` to an `XMLTYPE`
GETCONTENTXMLTYPE Function	Retrieves the contents of a resource returned as an `XMLTYPE`
GETLOCKTOKEN Procedure	Returns that resource's lock token for the current user given a path to a resource
GETPRIVILEGES Function	Gets all privileges granted to the current user on a specified resource
GETRESOID Function	Returns the object ID of the resource from its absolute path
GETXDB_TABLESPACE Function	Returns the current tablespace of the XDB (user)
HASBLOBCONTENT Function	Returns `TRUE` if the resource has BLOB content
HASCHARCONTENT Function	Returns `TRUE` if the resource has character content
HASXMLCONTENT Function	Returns `TRUE` if the resource has XML content
HASXMLREFERENCE Function	Returns `TRUE` if the resource has `REF` to XML content
ISFOLDER Function	Returns `TRUE` if the resource is a folder or container
LINK Procedures	Creates a link to an existing resource
LOCKRESOURCE Function	Gets a WebDAV-style lock on that resource given a path to that resource
PROCESSLINKS Procedure	Processes document links in the specified resource
PURGERESOURCEMETADATA Procedure	Deletes all user metadata from a resource
RENAMERESOURCE Procedure	Renames the XDB resource
SETACL Procedure	Sets the ACL on a specified resource
SPLITPATH Procedure	Splits the path into a parentpath and childpath
TOUCHRESOURCE Procedure	Changes the modification time of the resource to the current time
UNLOCKRESOURCE Function	Unlocks the resource given a lock token and resource path
UPDATERESOURCEMETADATA Procedures	Updates metadata for a resource

ACLCHECKPRIVILEGES Function

This function checks access privileges granted to the current user by specified ACL document by the `OWNER` of the resource. Returns positive integer if all privileges are granted.

Syntax

```
DBMS_XDB_REPOS.ACLCHECKPRIVILEGES(
   acl_path  IN  VARCHAR2,
   owner     IN  VARCHAR2,
   privs     IN  xmltype)
 RETURN PLS_INTEGER;
```

Parameters

Table 228-3 *ACLCHECKPRIVILEGES Function Parameters*

Parameter	Description
`acl_path`	Absolute path in the Hierarchy for ACL document
`owner`	Resource owner name; the pseudo user "DAV:owner" is replaced by this user during ACL privilege resolution
`privs`	An `XMLType` instance of the privilege element specifying the requested set of access privileges. See description for CHECKPRIVILEGES Function.

APPENDRESOURCEMETADATA Procedure

This procedure takes in user-defined metadata either as a `REF` to `XMLTYPE` or an `XMLTYPE` and adds it to the desired resource.

Syntax

```
DBMS_XDB_REPOS.APPENDRESOURCEMETADATA (
 abspath   IN  VARCHAR2,
 metadata  IN  XMLTYPE);

DBMS_XDB_REPOS.APPENDRESOURCEMETADATA (
 abspath   IN  VARCHAR2,
 metadata  IN  REF SYS.XMLTYPE);
```

Parameters

Table 228-4 APPENDRESOURCEMETADATA Procedure

Parameter	Description
`abspath`	Absolute path of the resource
`metadata`	Metadata can be schema based or nonschema-based. Schema-based metadata is stored in its own table.

Usage Notes

- In the case in which a `REF` is passed in, the procedure stores the `REF` in the resource, and the metadata is stored in a separate table. In this case you are responsible for populating the `RESID` column for the metadata table. Note that the`REF` passed in must be unique. In

other words, there must not be a`REF` with the same value in the resource metadata, as this would violate uniqueness of properties. An error is thrown if users attempt to add a `REF` that already exists.

- In the case where the XMLTYPE is passed in, the data is parsed to determine if it is schema-based or not and stored accordingly.

CHANGEOWNER Procedure

This procedure changes the owner of the resource/s to the specified owner.

Syntax

```
DBMS_XDB_REPOS.CHANGEOWNER(
     abspath    IN   VARCHAR2,
     owner      IN   VARCHAR2,
     recurse    IN   BOOLEAN := FALSE);
```

Parameters

Table 228-5 CHANGEOWNER Procedure Parameters

Parameter	Description
`abspath`	Absolute path of the resource
`owner`	New owner for the resource
`recurse`	If `TRUE`, recursively change owner of all resources in the folder tree

CHANGEPRIVILEGES Function

This function adds a specified ACE to a specified resource's ACL.

Syntax

```
DBMS_XDB_REPOS.CHANGEPRIVILEGES(
   res_path   IN    VARCHAR2,
   ace        IN    xmltype)
 RETURN PLS_INTEGER;
```

Parameters

Table 228-6 CHANGEPRIVILEGES Function Parameters

Parameter	Description
`res_path`	Path name of the resource for which privileges need to be changed
`ace`	An `XMLType` instance of the `<ace>` element which specifies the `<principal>`, the operation `<grant>` and the list of privileges

Return Values

A positive integer if the ACL was successfully modified.

Usage Notes

If no ACE with the same principal and the same operation (`grant`/`deny`) already exists in the ACL, the new ACE is added at the end of the ACL.

CHECKPRIVILEGES Function

This function checks access privileges granted to the current user on the specified resource.

Syntax

```
DBMS_XDB_REPOS.CHECKPRIVILEGES(
   res_path   IN  VARCHAR2,
   privs      IN  xmltype)
 RETURN PLS_INTEGER;
```

Parameters

Table 228-7 CHECKPRIVILEGES Function Parameters

Parameter	Description
`res_path`	Absolute path in the Hierarchy for resource
`privs`	An `XMLType` instance of the privilege element specifying the requested set of access privileges

Return Values

A positive integer if all requested privileges granted.

CREATEFOLDER Function

This function creates a new folder resource in the hierarchy.

Syntax

```
DBMS_XDB_REPOS.CREATEFOLDER(
   path   IN  VARCHAR2)
 RETURN BOOLEAN;
```

Parameters

Table 228-8 CREATEFOLDER Function Parameters

Parameter	Description
`path`	Path name for the new folder

Return Values

`TRUE` if operation successful; `FALSE`, otherwise.

Usage Notes

The given path name's parent folder must already exist in the hierarchy: if `'/folder1/folder2'` is passed as the path parameter, then `'/folder1'` must already exist.

CREATEOIDPATH Function

This function creates a virtual path to the resource based on object ID.

Syntax

```
DBMS_XDB_REPOS.CREATEOIDPATH(
   oid    IN   RAW)
 RETURN VARCHAR2;
```

Parameters

Table 228-9 CREATEOIDPATH Function Parameters

Parameter	Description
`oid`	Object ID of the resource

CREATERESOURCE Functions

The functions create a new resource. The description of the overload options precede each version of the syntax

Syntax

Creates a new resource with a specified string as its contents:

```
DBMS_XDB_REPOS.CREATERESOURCE(
     abspath        IN  VARCHAR2,
     data           IN  VARCHAR2)
 RETURN BOOLEAN;
```

Creates a new resource with a specified `XMLType` data as its contents:

```
DBMS_XDB_REPOS.CREATERESOURCE(
     abspath        IN  VARCHAR2,
     data           IN  SYS.XMLTYPE)
 RETURN BOOLEAN;
```

Given a `REF` to an existing `XMLType` row, creates a resource whose contents point to that row. That row should not already exist inside another resource:

```
DBMS_XDB_REPOS.CREATERESOURCE(
     abspath        IN  VARCHAR2,
     datarow        IN  REF SYS.XMLTYPE)
 RETURN BOOLEAN;
```

Creates a resource with a specified `BLOB` as its contents, and specifies character set of the source `BLOB`:

```
DBMS_XDB_REPOS.CREATERESOURCE(
     abspath        IN  VARCHAR2,
     data           IN  BLOB,
     csid           IN  NUMBER :=0)
 RETURN BOOLEAN;
```

Creates a resource with a specified `BFILE` as its contents, and specifies character set of the source `BFILE`:

```
DBMS_XDB_REPOS.CREATERESOURCE (
     abspath        IN  VARCHAR2,
     data           IN  BFILE,
     csid           IN  NUMBER :=0)
 RETURN BOOLEAN;
```

Creates a resource with a specified `CLOB` as its contents:

```
DBMS_XDB_REPOS.CREATERESOURCE (
     abspath        IN  VARCHAR2,
     data           IN  CLOB)
 RETURN BOOLEAN;
```

Given a string, inserts a new resource into the hierarchy with the string as the contents:

```
DBMS_XDB_REPOS.CREATERESOURCE (
   abspath        IN  VARCHAR2,
   data           IN  VARCHAR2,
   schemaurl      IN  VARCHAR2 := NULL,
   elem           IN  VARCHAR2 := NULL)
 RETURN BOOLEAN;
```

Given an `XMLTYPE` and a schema URL, inserts a new resource into the hierarchy with the `XMLTYPE` as the contents:

```
DBMS_XDB_REPOS.CREATERESOURCE (
   abspath        IN  VARCHAR2,
   data           IN  SYS.XMLTYPE,
   schemaurl      IN  VARCHAR2 := NULL,
   elem           IN  VARCHAR2 := NULL)
 RETURN BOOLEAN;
```

Parameters

Table 228-10 CREATERESOURCE Function Parameters

Parameter	Description
`abspath`	Absolute path of the resource to create. The path name's parent folder must already exist in the hierarchy. In other words, if `/foo/bar.txt` is passed in, then folder `/foo` must already exist.
`data`	String buffer containing new resource's contents. The data is parsed to check if it contains a schema-based XML document, and the contents are stored as schema-based in the schema's default table. Otherwise, it is saved as binary data.
`datarow`	`REF` to an `XMLType` row to be used as the contents
`csid`	Character set id of the document. Must be a valid Oracle ID; otherwise returns an error. If CSID is not specified, or if a zero CSID is specified, then the character set id of the document is determined as follows: • From the abspath extension, determine the resource's MIME type. • If the MIME type is */xml, then the encoding is detected based on Appendix F of the W3C XML 1.0 Reference at http://www.w3.org/TR/2000/REC-xml-20001006; • Otherwise, it is defaulted to the database character set.
`schemaurl`	For XML data, schema URL data conforms to (default `NULL`)
`elem`	Element name (default `NULL`)

Return Values

`TRUE` if operation successful; `FALSE`, otherwise.

DELETERESOURCE Procedure

This procedure deletes a resource from the hierarchy.

Syntax

```
DBMS_XDB_REPOS.DELETERESOURCE(
   path          IN      VARCHAR2,
   delete_option IN      PLS_INTEGER);
```

Parameters

Table 228-11 DELETERESOURCE Procedure Parameters

Parameter	Description
`path`	Path name of the resource to delete
`delete_option`	The option that controls how a a resource is deleted: • `DELETE_RESOURCE` • `DELETE_RECURSIVE` • `DELETE_FORCE` • `DELETE_RECURSIVE_FORCE`

DELETERESOURCEMETADATA Procedures

This procedure takes in a resource by absolute path and removes either the schema-based metadata identified by the REF, or the metadata identified by the namespace and name combination, which can be either schema-based or non-schema based. It also takes an additional (optional) parameter that specifies how to delete it. This parameter is only relevant for schema-based resource metadata that needs to be deleted. For non-schema based metadata, this parameter is ignored.

Syntax

Can be used only for schema-based metadata:

```
DBMS_XDB_REPOS.DELETERESOURCEMETADATA (
   abspath        IN  VARCHAR2,
   metadata       IN  REF SYS.XMLTYPE,
   delete_option  IN  pls_integer := DBMS_XDB_REPOS.DELETE_RESOURCE_METADATA_CASCADE);
```

Can be used for schema-based or nonschema-based metadata:

```
DBMS_XDB_REPOS.DELETERESOURCEMETADATA (
   abspath        IN VARCHAR2,
   metadatans     IN VARCHAR2,
   metadataname   IN VARCHAR2,
   delete_option  IN pls_integer := DBMS_XDB_REPOS.DELETE_RESOURCE_METADATA_CASCADE);
```

Parameters

Table 228-12 DELETERESOURCEMETADATA Procedure Parameters

Parameter	Description
abspath	Absolute path of the resource
metadata	REF to the piece of metadata (schema based) to be deleted
mettadatans	Namespace of the metadata fragment to be removed
mettadataname	Local name of the metadata fragment to be removed
delete_option	Only applicable for schema-based metadata, this can be one of the following: • DELETE_RES_METADATA_CASCADE - deletes the corresponding row in the metadata table • DELETE_RES_METADATA_NOCASCADE - does not delete the row in the metadata table

EXISTSRESOURCE Function

This function indicates if a resource is in the hierarchy. It matches the resource by a string that represents its absolute path.

Syntax

```
DBMS_XDB_REPOS.EXISTSRESOURCE(
   abspath    IN    VARCHAR2)
 RETURN BOOLEAN;
```

Parameters

Table 228-13 EXISTSRESOURCE Function Parameters

Parameter	Description
abspath	Path name of the resource whose ACL document is required

Return Values

TRUE if the resource is found.

GETACLDOCUMENT Function

This function retrieves ACL document that protects resource given its path name.

Syntax

```
DBMS_XDB_REPOS.GETACLDOCUMENT(
   abspath  IN  VARCHAR2)
  RETURN sys.xmltype;
```

Parameters

Table 228-14 GETACLDOCUMENT Function Parameters

Parameter	Description
abspath	Path name of the resource whose ACL document is required

Return Values

The XMLType for ACL document.

GETCONTENTBLOB Function

This function retrieves the contents of a resource returned as a BLOB.

Syntax

```
DBMS_XDB_REPOS.GETCONTENTBLOB(
     abspath    IN     VARCHAR2,
     csid       OUT    PLS_INTEGER,
     locksrc    IN     BOOLEAN := FALSE)
  RETURN BLOB;
```

Parameters

Table 228-15 GETCONTENTBLOB Function Parameters

Parameter	Description
abspath	Absolute path of the resource
csid	If TRUE, lock and return the source LOB. If FALSE, return a temp LOB copy.
locksrc	Contents of the resource as a BLOB

Return Values

The contents of the resource as a BLOB.

GETCONTENTCLOB Function

This function gets the contents of a resource returned as a CLOB.

Syntax

```
DBMS_XDB_REPOS.GETCONTENTCLOB(
     abspath    IN     VARCHAR2,
  RETURN CLOB;
```

Parameters

Table 228-16 GETCONTENTCLOB Function Parameters

Parameter	Description
abspath	Absolute path of the resource

Return Values

The contents of the resource as a CLOB.

GETCONTENTVARCHAR2 Function

This function gets the contents of a resource returned as a string.

Syntax

```
DBMS_XDB_REPOS.GETCONTENTVARCHAR2(
     abspath    IN     VARCHAR2,
  RETURN BLOB;
```

Parameters

Table 228-17 GETCONTENTVARCHAR2 Function Parameters

Parameter	Description
abspath	Absolute path of the resource

Return Values

The contents of the resource as a string.

GETCONTENTXMLREF Function

This function retrieves the contents of a resource returned as a a `REF` to an `XMLTYPE`.

Syntax

```
DBMS_XDB_REPOS.GETCONTENTXMLREF(
     abspath    IN     VARCHAR2,
  RETURN SYS.XMLTYPE;
```

Parameters

Table 228-18 GETCONTENTXMLREF Function Parameters

Parameter	Description
abspath	Absolute path of the resource

Return Values

The contents of the resource as a `REF` to an `XMLTYPE`.

GETCONTENTXMLTYPE Function

This function retrieves the contents of a resource returned as an `XMLTYPE`.

Syntax

```
DBMS_XDB_REPOS.GETCONTENTXMLTYPE(
     abspath     IN      VARCHAR2,
  RETURN SYS.XMLTYPE;
```

Parameters

Table 228-19 GETCONTENTXMLTYPE Function Parameters

Parameter	Description
`abspath`	Absolute path of the resource

Return Values

The contents of the resource as an XMLTYPE.

GETLOCKTOKEN Procedure

Given a path to a resource, this procedure returns that resource's lock token for the current user.

Syntax

```
DBMS_XDB_REPOS.GETLOCKTOKEN(
   path          IN      VARCHAR2,
   locktoken     OUT     VARCHAR2);
```

Parameters

Table 228-20 GETLOCKTOKEN Procedure Parameters

Parameter	Description
`path`	Path name to the resource
`locktoken`	Logged-in user's lock token for the resource

Usage Notes

The user must have `READPROPERTIES` privilege on the resource.

GETPRIVILEGES Function

This function gets all privileges granted to the current user on a specified resource.

Syntax

```
DBMS_XDB_REPOS.GETPRIVILEGES(
   res_path     IN      VARCHAR2)
 RETURN sys.xmltype;
```

Parameters

Table 228-21 GETPRIVILEGES Function Parameters

Parameter	Description
res_path	Absolute path in the hierarchy of the resource

Return Values

An XMLType instance of <privilege> element, which contains the list of all leaf privileges granted on this resource to the current user.

GETRESOID Function

The GETRESOID function returns the object ID of the resource from its absolute path.

Syntax

```
DBMS_XDB_REPOS.GETRESOID(
   abspath IN VARCHAR2)
RETURN RAW;
```

Parameters

Table 228-22 GETRESOID Function Parameters

Parameter	Description
abspath_path	Absolute path of the resource

Return Values

NULL if the resource is not present.

GETXDB_TABLESPACE Function

This function returns the current tablespace of the XDB (user).

Syntax

```
DBMS_XDB_REPOS.GETXDB_TABLESPACE
 RETURN VARCHAR2;
```

HASBLOBCONTENT Function

This function returns TRUE if the resource has BLOB content.

Syntax

```
DBMS_XDB_REPOS.HASBLOBCONTENT
     abspath    IN     VARCHAR2)
 RETURN BOOLEAN;
```

Parameters

Table 228-23 HASBLOBCONTENT Function Parameters

Parameter	Description
abspath_path	Absolute path of the resource

Return Values

TRUE if the resource has BOB content.

HASCHARCONTENT Function

This function returns TRUE if the resource has character content.

Syntax

```
DBMS_XDB_REPOS.HASCHARCONTENT
     abspath    IN     VARCHAR2)
 RETURN BOOLEAN;
```

Parameters

Table 228-24 HASCHARCONTENT Function Parameters

Parameter	Description
abspath_path	Absolute path of the resource

Return Values

TRUE if the resource has character content.

HASXMLCONTENT Function

This function returns TRUE if the resource has XML content.

Syntax

```
DBMS_XDB_REPOS.HASXMLCONTENT
     abspath    IN     VARCHAR2)
 RETURN BOOLEAN;
```

Parameters

Table 228-25 HASXMLCONTENT Function Parameters

Parameter	Description
abspath_path	Absolute path of the resource

Return Values

TRUE if the resource has XML content.

HASXMLREFERENCE Function

This function returns `TRUE` if the resource has a `REF` to XML content.

Syntax

```
DBMS_XDB_REPOS.HASXMLREFERENCE
     abspath     IN      VARCHAR2)
 RETURN BOOLEAN;
```

Parameters

Table 228-26 HASXMLREFERENCE Function Parameters

Parameter	Description
`abspath_path`	Absolute path of the resource

Return Values

`TRUE` resource has a `REF` to XML content.

ISFOLDER Function

This function returns `TRUE` if the resource is a folder or container.

Syntax

```
DBMS_XDB_REPOS.ISFOLDER
     abspath     IN      VARCHAR2)
 RETURN BOOLEAN;
```

Parameters

Table 228-27 ISFOLDER Function Parameters

Parameter	Description
`abspath_path`	Absolute path of the resource

Return Values

`TRUE` if the resource is a folder or container.

LINK Procedures

This procedure creates from a specified folder to a specified resource.

Syntax

```
DBMS_XDB_REPOS.LINK(
   srcpath      IN   VARCHAR2,
   linkfolder   IN   VARCHAR2,
   linkname     IN   VARCHAR2);
```

```
DBMS_XDB_REPOS.LINK(
   srcpath      IN   VARCHAR2,
   linkfolder   IN   VARCHAR2,
   linkname     IN   VARCHAR2,
   linktype     IN   PLS_INTEGER := DBMS_XDB_REPOS.LINK_TYPE_HARD);
```

Parameters

Table 228-28 LINK Procedure Parameters

Parameter	Description
`srcpath`	Path name of the resource to which a link is created
`linkfolder`	Folder in which the new link is placed
`linkname`	Name of the new link
`linktype`	Type of link to be created: • `DBMS_XDB.LINK_TYPE_HARD` (default) • `DBMS_XDB.LINK_TYPE_WEAK` • `DBMS_XDB.LINK_TYPE_SYMBOLIC`

LOCKRESOURCE Function

Given a path to a resource, this function gets a WebDAV-style lock on that resource.

Syntax

```
DBMS_XDB_REPOS.LOCKRESOURCE(
   path      IN  VARCHAR2,
   depthzero IN  BOOLEAN,
   shared    IN  boolean)
RETURN BOOLEAN;
```

Parameters

Table 228-29 LOCKRESOURCE Function Parameters

Parameter	Description
`path`	Path name of the resource to lock.
`depthzero`	Currently not supported
`shared`	Passing `TRUE` obtains a shared write lock

Return Values

`TRUE` if successful.

Usage Notes

The user must have `UPDATE` privileges on the resource.

PROCESSLINKS Procedure

This procedure processes document links in the specified resource.

Syntax

```
DBMS_XDB_REPOS.PURGERESOURCEMETADATA(
 abspath  IN  VARCHAR2,
 recurse  IN  BOOLEAN := FALSE);
```

Parameters

Table 228-30 PROCESSLINKS Procedure Parameters

Parameter	Description
`abspath`	Absolute path of the resource. If the path is a folder, use the `recurse` flag.
`recurse`	Used only if `abspath` specifies a folder. If `TRUE`, process links of all resources in the folder hierarchy rooted at the specified resource. If `FALSE`, process links of all documents in this folder only.

PURGERESOURCEMETADATA Procedure

This procedure deletes all user metadata from a resource. Schema-based metadata is removed in cascade mode, rows being deleted from the corresponding metadata tables.

Syntax

```
DBMS_XDB_REPOS.PURGERESOURCEMETADATA(
 abspath  IN  VARCHAR2);
```

Parameters

Table 228-31 PURGERESOURCEMETADATA Procedure Parameters

Parameter	Description
`abspath`	Absolute path of the resource

RENAMERESOURCE Procedure

This procedure renames the XDB resource.

Syntax

```
DBMS_XDB_REPOS.RENAMERESOURCE(
   srcpath    IN  VARCHAR2,
   destfolder IN  CARCHAR2,
   newname    IN  VARCHAR2);
```

Parameters

Table 228-32 RENAMERESOURCE Procedure Parameters

Parameter	Description
srcpath	Absolute path in the Hierarchy for the source resource destination folder
destfolder	Absolute path in the Hierarchy for the destination folder
newname	Name of the child in the destination folder

SETACL Procedure

This procedure sets the ACL on a specified resource to be the ACL specified by path.

Syntax

```
DBMS_XDB_REPOS.SETACL(
   res_path   IN  VARCHAR2,
   acl_path   IN  VARCHAR2);
```

Parameters

Table 228-33 SETACL Procedure Parameters

Parameter	Description
res_path	Absolute path in the Hierarchy for resource
acl_path	Absolute path in the Hierarchy for ACL

Usage Notes

The user must have <write-acl> privileges on the resource.

SPLITPATH Procedure

This procedure splits the path into a parentpath and childpath.

Syntax

```
DBMS_XDB_REPOS.SPLITPATH(
      abspath     IN  VARCHAR2,
      parentpath  OUT VARCHAR2,
     childpath    OUT VARCHAR2);
```

Parameters

Table 228-34 SPLITPATH Procedure Parameters

Parameter	Description
abspath	Absolute path to be split
parentpath	Parentpath
childpath	Childpath

TOUCHRESOURCE Procedure

This procedure changes the modification time of the resource to the current time.

Syntax

```
DBMS_XDB_REPOS.TOUCHRESOURCE
     abspath     IN      VARCHAR2);
```

Parameters

Table 228-35 TOUCHRESOURCE Procedure Parameters

Parameter	Description
abspath_path	Absolute path of the resource

UNLOCKRESOURCE Function

This function unlocks the resource given a lock token and a path to the resource.

Syntax

```
DBMS_XDB_REPOS.UNLOCKRESOURCE(
   path     IN  VARCHAR2,
   deltoken IN  VARCHAR2)
 RETURN BOOLEAN;
```

Parameters

Table 228-36 UNLOCKRESOURCE Function Parameters

Parameter	Description
path	Path name to the resource
deltoken	Lock token to be removed

Return Values

TRUE if operation successful.

Usage Notes

The user must have UPDATE privileges on the resource.

UPDATERESOURCEMETADATA Procedures

This procedure updates metadata for a resource.

The procedure takes in a resource identified by absolute path and the metadata in it to replace identified by its REF. It replaces that piece of metadata with user-defined metadata which is either in the form of a REF to XMLTYPE or an XMLTYPE.

Syntax

Can be used to update schema-based metadata only. The new metadata must be schema-based:

```
DBMS_XDB_REPOS.UPDATERESOURCEMETADATA(
   abspath  IN VARCHAR2,
   oldmetadata    IN   REF SYS.XMLTYPE,
   newmetadata    IN   REF SYS.XMLTYPE)
```

Can be used to update schema-based metadata only. The new metadata must be schema-based or nonschema-based:

```
DBMS_XDB_REPOS.UPDATERESOURCEMETADATA(
   abspath        IN   VARCHAR2,
   oldmetadata    IN   REF SYS.XMLTYPE,
   newmetadata    IN   XMLTYPE);
```

Can be used for both schema-based and nonschema-based metadata:

```
DBMS_XDB_REPOS.UPDATERESOURCEMETADATA(
   abspath        IN   VARCHAR2,
   oldns          IN   VARCHAR2,
   oldname IN VARCHAR,
   newmetadata IN XMLTYPE);
```

Can be used for both schema-based or nonschema-based metadata. New metadata must be schema-based:

```
DBMS_XDB_REPOS.UPDATERESOURCEMETADATA(
   abspath       IN   VARCHAR2,
   oldns         IN   VARCHAR2,
   oldname       IN   VARCHAR,
   newmetadata   IN   REF SYS.XMLTYPE);
```

Parameters

Table 228-37 UPDATERESOURCEMETADATA Procedure Parameters

Parameter	Description
`abspath`	Absolute path of the resource
`oldmetadata`	`REF` to the old of metadata
`newmetadata`	`REF` to the new, replacement metadata (can be either schema-based or nonschema-based depending on the overload)
`oldns`	Namespace identifying old metadata
`oldname`	Local name identifying old metadata

Usage Notes

In the case of `REF`, it stores the `REF` in the resource and the metadata is stored in a separate table. Uniqueness of `REFs` is enforced. In the case where the `XMLTYPE` is passed in, data is parsed to determine if it is schema-based or not and is stored accordingly.