

Configuration of Oracle XML DB Repository

Overall configuration of Oracle XML DB Repository applies to all repository resources. It does not include configuring parameters for handling events or managing XLink and XInclude processing. You use resource configuration files to configure resources.

**Note:**

The Oracle XML DB Repository is deprecated with Oracle Database 23ai.

- [Resource Configuration Files](#)
You configure an Oracle XML DB Repository resource for any purpose by associating it with a resource configuration file, which defines configurable parameters for the resource. A resource configuration file is itself a resource in Oracle XML DB Repository.
- [Configuring a Resource](#)
Configuring an Oracle XML DB Repository resource involves creating a configuration file, adding that file to the repository as a configuration resource, and mapping the configuration resource to the resources it configures or to the entire repository.
- [Common Configuration Parameters](#)
Commonly used configuration parameters are described, that is, elements in a configuration file.

Related Topics

- [Oracle XML DB Protocol Server Configuration Management](#)
Oracle XML DB protocol server uses configuration parameters stored in file `xdbconfig.xml` to initialize its startup state and manage session level configuration. The session pool size and timeout parameters cannot be changed dynamically, that is, you must restart the database in order for these changes to take effect.
- [Configuration of Repository Events](#)
In a resource configuration file, you use element `event-listeners`, child of element `ResConfig`, to configure Oracle XML DB Repository event handling.
- [Configuration of Repository Resources for XLink and XInclude](#)
The resource configuration file that you use as a resource to configure XLink (*deprecated*) and XInclude processing for other resources is described.
- [XDBResource.xsd: XML Schema for Oracle XML DB Resources](#)
A full listing is presented of the Oracle XML DB-supplied XML schema `XDBResource.xsd`, which is used to represent Oracle XML DB resources.
- [XDBResConfig.xsd: XML Schema for Resource Configuration](#)
A full listing is presented of the Oracle XML DB-supplied XML schema used to configure repository resources. It is accessible in Oracle XML DB Repository at path `/sys/schemas/PUBLIC/xmlns.oracle.com/xdb/XDBResConfig.xsd`.

**See Also:**

[Package DBMS_XDB_ADMIN](#), for information about using a dedicated tablespace for the repository

Resource Configuration Files

You configure an Oracle XML DB Repository resource for any purpose by associating it with a resource configuration file, which defines configurable parameters for the resource. A resource configuration file is itself a resource in Oracle XML DB Repository.

A **resource configuration file** is an XML file that conforms to the XML schema `XDBResConfig.xsd`, which is accessible in Oracle XML DB Repository at path `/sys/schemas/PUBLIC/xmlns.oracle.com/xdb/XDBResConfig.xsd`. This XML schema is defined by Oracle XML DB, and you cannot alter it.

You use PL/SQL procedure `DBMS_RESCONFIG.addResConfig` to map a resource to the file that configures it. A single resource configuration file can alternatively apply to all resources in the repository. In that case, you use PL/SQL procedure `DBMS_RESCONFIG.addRepositoryResConfig` to map it to the repository as a whole.

The same resource configuration file can be used to configure more than one resource, if appropriate. Oracle recommends that you have resources share a configuration file this way whenever the same configuration makes sense. This can improve run-time performance. It also simplifies repository management by letting you update a configuration in a single place and have the change affect multiple resources.

Avoid creating multiple, equivalent resource configuration files, because that can impact performance negatively. If Oracle XML DB detects duplicate resource configuration files, it raises an error.

Typically, you configure a resource for use with a particular application. In order for a resource to be shared by multiple applications, it must be possible for different applications to configure it differently. You do this by creating multiple resource configuration files and mapping them to the same resource. Each resource is thus associated with a list of configurations, a **resource configuration list**. Configurations in a configuration list are processed in the list order.

The repository itself has a list of resource configuration files, for repository-wide configuration, which really means configuration of all resources in the repository. The same configuration file must not be used for both the repository itself and a specific resource. Otherwise, an error is raised. An error is also raised if the same resource configuration file appears more than once in any given resource configuration list.

**Note:**

An error is raised if you try to create more than 125 resource configuration files for repository-wide configuration.

The resource configuration list of a new resource is based on the information in the `configuration` elements of all resource configuration files for the parent folder of the new resource. If there is no such information (no configuration file or no `defaultChildConfig` elements in the files), then the `configuration` elements of the repository resource

configuration list are used. If that information is also missing, then the new resource has an empty resource configuration list.

You can view the configuration list for a particular resource by extracting element `/Resource/RCList` from column `RES` of view `RESOURCE_VIEW`, or by using PL/SQL procedure `DBMS_RESCONFIG.getResConfigPath`. You can view the configuration list for the repository as a whole by using PL/SQL procedure `DBMS_RESCONFIG.getRepositoryResConfigPath`. To modify the repository-wide configuration list, you must be granted role `XDBADMIN`.

Related Topics

- [Configuration Elements `defaultChildConfig` and `configuration`](#)
Configuration element `defaultChildConfig` applies to only folders. It holds configuration information to be applied to all child resources in the folder. Element `defaultChildConfig` has one or more `configuration` child elements, each of which defines a possible configuration for resources in the folder.

Configuring a Resource

Configuring an Oracle XML DB Repository resource involves creating a configuration file, adding that file to the repository as a configuration resource, and mapping the configuration resource to the resources it configures or to the entire repository.

Follow these steps to configure an individual resource or the repository as a whole (all resources):

1. Create a resource configuration file that defines the configuration. This XML file must conform to XML schema `XDBResConfig.xsd`.
2. Add the resource configuration file to the repository as a resource in its own right: a configuration resource. You can use PL/SQL function `DBMS_XDB_REPOS.createResource` to do this.
3. Map this configuration resource to the resources that it configures, or to the repository if it applies to all resources. Use PL/SQL procedure `DBMS_RESCONFIG.addResConfig` or `DBMS_RESCONFIG.appendResConfig` to map an individual resource. Use `DBMS_RESCONFIG.addRepositoryResConfig` to map the repository as a whole.
4. Commit.



Note:

Before performing any operation that uses a resource configuration file, you must perform a `COMMIT` operation. Until you do that, an ORA-22881 "dangling REF" error is raised whenever you use the configuration file.

PL/SQL package `DBMS_RESCONFIG` provides additional procedures to delete a configuration from a configuration list, obtain a list of paths to configurations in a configuration list, and more.

**Note:**

If you delete a resource configuration file that is referenced by another resource, a dangling REF error is raised whenever an attempt is made to access the configured resource.

Related Topics

- [XDBResConfig.xsd: XML Schema for Resource Configuration](#)
A full listing is presented of the Oracle XML DB-supplied XML schema used to configure repository resources. It is accessible in Oracle XML DB Repository at path `/sys/schemas/PUBLIC/xmlns.oracle.com/xdb/XDBResConfig.xsd`.

**See Also:**

- [Example 22-1](#) for an example of a simple resource configuration file
- *Oracle Database PL/SQL Packages and Types Reference* for information about package `DBMS_RESCONFIG`

Common Configuration Parameters

Commonly used configuration parameters are described, that is, elements in a configuration file.

Parameters specific to particular types of configuration are described elsewhere.

- [Configuration Element ResConfig](#)
The top-level element of a resource configuration file is `ResConfig`. You can use it to disable or enable the resource configuration file.
- [Configuration Elements defaultChildConfig and configuration](#)
Configuration element `defaultChildConfig` applies to only folders. It holds configuration information to be applied to all child resources in the folder. Element `defaultChildConfig` has one or more `configuration` child elements, each of which defines a possible configuration for resources in the folder.
- [Configuration Element applicationData](#)
Element `applicationData` stores application-specific data. An application typically passes this data to an event handler when the handler is run. You can use any XML content you want inside element `applicationData`.

Configuration Element ResConfig

The top-level element of a resource configuration file is `ResConfig`. You can use it to disable or enable the resource configuration file.

Besides attributes `namespace` and `schemaLocation`, element `ResConfig` can contain an optional `enable` attribute. Set the value of attribute `enable` to `false` to disable the resource configuration file, so that it has no effect on the resources mapped to it. This can be useful for

debugging or disabling an application. The default value of `enable`, used if the attribute is not present, is `true`.

Configuration Elements `defaultChildConfig` and `configuration`

Configuration element `defaultChildConfig` applies to only folders. It holds configuration information to be applied to all child resources in the folder. Element `defaultChildConfig` has one or more `configuration` child elements, each of which defines a possible configuration for resources in the folder.

A `configuration` element has the following child elements:

- `pre-condition` (optional) – This element specifies a condition that must be met before the resource configuration identified by the `path` element (see next) can be used as the default configuration. If element `pre-condition` is absent, then the resource configuration file targeted by `path` applies to all resources in the folder. That is, the precondition is treated as `true`.

A `pre-condition` element has an optional `existsNode` child element. An `existsNode` element has a required `XPath` child element and an optional `namespace` child element, both strings. These define an XPath 1.0 expression and a namespace, respectively, that are used to check the existence of a resource. If that resource exists, then the precondition is satisfied, so the resource configuration file identified by `path` is used as a default resource configuration file for all child resources in the folder. The first component of the `XPath` element must be `Resource`.

Note:

A complex XPath expression for element `XPath` can impact performance negatively.

If multiple `configuration` elements have true preconditions, then each of the resource configuration files identified by their associated `path` elements applies to all of the resources in the folder.

- `path` (required) – This element specifies an absolute repository path to a resource configuration file that is to be used as the default configuration for a new resource whenever the precondition specified by element `pre-condition` is satisfied.

Typically, the value of the `path` element is a path to the current resource configuration file, that is, the file that contains the `path` element. [Example 22-1](#) illustrates this, assuming that the resource configuration file is located at path `/cm/app_rc.xml` in the repository. In this example, the precondition is that there be a `Resource` node whose content is of type `xml`. When that precondition is met, the resource configuration file in [Example 22-1](#) applies to all resources in same folder as the configuration file (`/cm/app_rc.xml`).

Example 22-1 Resource Configuration File

```
<ResConfig xmlns="http://xmlns.oracle.com/xdb/XDBResConfig.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.oracle.com/xdb/XDBResConfig.xsd
    http://xmlns.oracle.com/xdb/XDBResConfig.xsd">
  <defaultChildConfig>
    <configuration>
```

```

    <pre-condition>
      <existsNode>
        <XPath>/Resource[ContentType="xml"]</XPath>
      </existsNode>
    </pre-condition>
    <path>/cm/app_rc.xml<path>
  </configuration>
</defaultChildConfig>
</ResConfig>

```

Configuration Element applicationData

Element `applicationData` stores application-specific data. An application typically passes this data to an event handler when the handler is run. You can use any XML content you want inside element `applicationData`.

An event handler uses PL/SQL function `DBMS_XEVENT.getApplicationData` or Java function `oracle.xdb.XMLType.getApplicationData` to access the data in the `applicationData` of the resource configuration file for the event listener.

The use of repository events to trigger application actions is deprecated in Oracle Database 21c (21.3). There is no replacement.

[Example 22-2](#) shows an `applicationData` element for use with an Oracle Spatial and Graph application.



See Also:

- *Oracle Database XML Java API Reference*, class `XDBRepositoryEvent` for information about Java function `oracle.xdb.XMLType.getApplicationData`
- [Example 30-1](#) for an example of a resource configuration file for event listeners

Example 22-2 applicationData Element

```

<applicationData>
  <spatial:data xmlns:spatial="http://oracle/cartridge/spatial.xsd">
    <spatial:xpos>5</spatial:xpos>
    <spatial:ypos>10</spatial:ypos>
  </spatial:data>
</applicationData>

```