DOML tag reference

Table of Contents

Chapter 1. Tag reference

This chapter contains an alphabetical reference of all the tags allowed in DOML files. Each entry corresponds to an XML tag, and contains the subsections:

- Content - tags that the tag can contain.
- Attributes - attributes the tag can have.
- Context - tags within which the tag can appear, in other words, the tags that can contain it.

So, for a tag <sampleTag>, whose attributes are attribute1, attribute2, and so on, whose context is <contextTag>, and which can contain contentTag, its general syntax would look like:

```
<contextTag>
  <sampleTag attribute1 attribute2 ...>
    <contentTag/>
  </sampleTag>
</contextTag>
```

<doml>

Root element of DOML files. This tag contains a database hierarchy that contains all the packages.

Content: <database>

Attributes: None

Context: None

<database>

Contains the package hierarchy, and specifies the database.

Content: <package>

Attributes:

1. database - The database attribute specifies the database vendor. The valid types are as follows:

Standard - Generated SQL code will conform to standard JDBC SQL. This is the default attribute.

Oracle - DODS will generate SQL optimized for Oracle databases.

Informix - DODS will generate SQL optimized for Informix databases.

MSQL - DODS will generate SQL optimized for MSQL databases.

Sybase - DODS will generate SQL optimized for Sybase databases.

PostgreSQL - DODS will generate SQL optimized for PostgreSQL databases.

DB2 - DODS will generate SQL optimized for DB2 databases.

QED - DODS will generate SQL optimized for QED databases.

2. templateset - Template set that will be used for java code generation. The possible values for templateset are:

- standard (default)
- multidb
- webdocwf
- multidb_webdocwf
- <any user defined template>

Context: <doml>

<package>

Each package can contain a sub-package or a table structure.

Content: <package>, <table>

Attributes:

1. id - The name of the package. The format for the name includes the parent package's id value. For example, if I had a package myPackage, and a sub-package of it called mySubPackage, mySubPackage's id value would be myPackage.mySubPackage.

Context: <database>

<table>

<table> describes a table in a database.

Content:<column>

Attributes:

1. id - Similar to the id attribute in <package>, <table>'s id contains the value of the table name located in the package. For example, if I had a package myPackage, a subpackage mySubPackage, and a table myTable, the id value is myPackage.mySubPackage.myTable.

2. DbTableName - The actual SQL table name. By default this is the same as the id value, minus the package information. For example, myPackage.mySubPackage.myTable's dbTableName is myTable.

3. IsFinal - The possible values for isFinal are:

- true
- false

4. isLazyLoading - This attribute specifies whether the DO will use lazy loading. If a DO uses lazy loading, when you supply a known ObjectId to create a DO instance, the DO instance is created but the corresponding row in the table is not retrieved until the first get() or set() method call is made. It delays the hit on the database until the moment the data is actually needed. The possible values for isLazyLoading are:

- true

- false

5. caching - This attribute specifies whether the DO is fully cached, partially cached, lru cached, or not cached. Cached DOs are stored locally, and queries for the cached DO do not go to the database table. The query instead looks up the information in the cache. The possible values for caching are:

- none (default)
- partial
- lru
- full

6. IsIndex - This attribute specifies whether the data object is index. The possible values for isIndex are:

- true
- false (default)

7. IsAbstract - This attribute specifies whether the data object is abstract. In DODS, all data objects that are extended must be abstract. This is because of how DODS handles database tables. Only non-abstract leaf classes have tables in the database. The possible values for isAbstract are:

- true
- false (default)

8. isEJB - This attribute is not currently used by DODS.

9. IsView - This attribute is not currently used by DODS.

10. ExtensionOf - The name of another table of which this table is an extension. The value of extensionOf must match the id of a previously defined table.

11. notUsingOid - Specifies whether the table is oid based.

Context: <package>

<column>

<column> describes a column in a database table.

Content: <error>, <javadoc>, <referenceObject>, <type>, <initialValue>

Attributes:

1. id - The name of the column in the database table.

2. usedForQuery - Specifies whether the values of the column will be used for queries. The possible values for usedForQuery are:

- true
- false

3. isConstant - Specifies whether the column contains a constant value. The possible values for isConstant are:

- true
- false

4. isPrimaryKey - This attribute specifies whether the table column will be a primary key. If isPrimaryKey is true, a primary key is created on this column in the table. The possible values for isPrimaryKey are:

- true
- false

5. isUnique - Specifies whether the column must contain unique values. The possible values for isUnique are:

- true
- false

Context: <table>

<javadoc>

The <javadoc> tag contains the text for Javadoc entries for the column.

Content: None

Attributes: None

Context: <column>

<referenceObject>

If the column is a reference to another table, <referenceObject> specifies the table.

Content: None

Attributes:

1. Constraint - Specifies whether the specified table row must exist. The possible values for constraint are:

- true
- false

2. foreignKeyColumn - Specifies the column in the referenced table.

3. ForeignKeyGroup - Specifies the group of the foreign keys.

4. Reference - Specifies the ID of the referenced table.

Context: <column>

<type>

<type> dictates the form of the data stored in the column. If no <type> is specified, the column contains all default values.

Content: None

Attributes:

1. Size - Specifies the size of data types that are commonly measured in width, like VARCHAR. Size must be an integer.

2. CanBeNull - Specifies whether the column can contain null values. The possible values for canBeNull are:

- true
- false

3. dbType - Specifies the internal SQL data type the database will use for this column. The default value of dbType is VARCHAR.

4. JavaType - Specifies the Java data type returned by the DO to the user when querying this attribute of the DO. The default value of javaType is String.

Context: <column>

<initialValue>

<initialValue>is used to specify a default initial value for the column.

Content: None

Attributes: None

Context: <column>

<index>

<index> is used to specify index columns.

Content: <indexColumn>

Attributes:

1. id - The name of the index constraint in the database table.

2. Unique - Specifies whether the index constraint is unique. The possible values for unique are:

- true
- false

3. clustered - Specifies whether the index constraint is clustered. The possible values for clustered are:

- true
- false

Context: <table>

<indexColumn>

<indexColumn> is used to specify each index column in the constraint index.

Content:None

Attributes:

1. id - The name of the column in the database table.

Context: <index>