

---

# Reladomo Object XML Generator

26 June 2006

## Table of Contents

1. Introduction .....	1
2. Usage .....	1
2.1. Ant Build Task .....	1
2.2. Caveats .....	2

## 1. Introduction

Reladomo maps database tables to Java objects. These objects are generated using a set of configuration and object definition XML (mapping) files created by the user. One must create one XML file for every Reladomo object. The object XML file is the metadata definition of the Reladomo object.

To expedite the creation of object XML files from existing schema, an object XML file generator has been created. It connects directly to a database, retrieving a list of the existing tables and generating object XML files that appropriately map to these tables. The generator improves Reladomo's usability and also ensures better consistency between the Reladomo objects' definitions and the tables to which they are mapped.

## 2. Usage

### 2.1. Ant Build Task

An Ant task provides users with a simple way to integrate XML generation into a build. Consider the following example as it points to a number of XML elements used by MithraDbDefinitionGenerator:

```
<target name="generate-para-mithra-xml-url"
  depends="compile-mithra, compile-mithra-generator">
  <taskdef name="mithra-gen-xml"
    classname="com.gs.fw.common.mithra.generator.
      objectxmlgenerator.MithraObjectXmlGenerator"
    loaderRef="mithraGenerator">
    <classpath refid="mithragen.classpath"/>
  </taskdef>
  <mithra-gen-xml userName="user"
    password="password"
    schema="schema"
    databaseType="sybase"
    driver="com.sybase.jdbc4.jdbc.SybDriver"
    url="jdbc:sybase:Tds:somehost.example.com:8111/
      schema"
    includeTables="TABLE_1, TABLE_2"
    outputDir="${para.home}/xml/mithra/generated"/>
</target>

<target name="generate-para-mithra-xml-ldap"
  depends="compile-mithra, compile-mithra-generator">
```

```
<taskdef name="mithra-gen-xml"
  classname="com.gs.fw.common.mithra.generator.
    objectxmlgenerator.MithraObjectXmlGenerator"
  loaderRef="mithraGenerator">
  <classpath refid="mithragen.classpath"/>
</taskdef>
<mithra-gen-xml userName="user"
  password="password"
  schema="schema"
  databaseType="sybase"
  ldapName="SOME_LDAP_NAME"
  includeTablesFromFile="include.txt"
  outputDir="{para.home}/xml/mithra/generated"/>
</target>
```

Of interest are the elements for the `mithra-gen-xml` attributes above.

The `userName` and `password` elements specify the login credentials for the database to be used. The `databaseType` element sets the database type to be used. Again, acceptable values for the type are "sybase" and "udb82". `outputDir` is the directory in which the XML files will be generated. All of these elements must be included.

The XML generator may connect to a database using a specified URL and JDBC driver or an LDAP name. The `driver` element specifies the driver to use when connecting to the database. The `url` is the URL of the database. It is typically prefixed with `jdbc` to establish the protocol. The `ldapName` is the name of LDAP server to which the generator is to connect.

The `schema` element is optional. In Sybase, the generator's schema is the equivalent of the catalog in JDBC. In UDB, it is the equivalent of the schema in JDBC.

The `includeTables` and `includeTablesFromFile` elements are optional. If not included, the generator will run on all tables present in the database (given the default schema or schema provided). `includeTables` is a comma-separated list of tables from which XML files should be generated. `includeTablesFromFile` is the URL of a file containing a newline-delimited list of tables from which XML files should be generated. `includeTables` and `includeTablesFromFile` should be used exclusively.

## 2.2. Caveats

Note that the generator currently accepts "sybase" and "udb82" as possible database types. If no database type is specified or an invalid type is provided, the generator throws a `BuildException`.

If no primary key is specified in a table, the generator will set the primary key in the object XML file to be the first unique index retrieved from the database. This may not be the intended primary key. To avoid such a problem, users should be careful to specifically set the primary key for every table.