Schema Synchronization with DataMgr

Posted At : July 13, 2007 5:34 PM | Posted By : Steve
Related Categories: DataMgr

One common problem that many developers face is Database Schema Synchronization, that is making sure that your database has the right schema (table and columns and whatnot). This is important when moving code from development to production (to avoid a long task list of columns and tables to add) and important for distributed programs (things like **BlogCFC** that are used in lots of places).

Typically, this is done by writing several SQL scripts the keep the database schema up to date. Then you run the correct script for the next upgrade. **Barney Boisvert** wrote is **SchemaTool** to make this easier.

The problem is compounded if you want to support multiple databases. Ideally the SQL syntax should be the same from database to database, but usually it isn't. This means that you also have write separate SQL scripts for each database that you support.

DataMgr solves all of these problems in a manner that I feel is rather elegant.

Before I discuss it, I should discus the limitations of the DataMgr approach. It only deals with tables and columns - no indexes, views, triggers, etc. It is only additive - it will add missing tables/columns, but it won't remove ones that you no longer need.

Now that I have that out of the way, here is how DataMgr can do schema synchronization:

DataMgr can match a schema from an XML document representing the database. Fortunately, you don't have to write the XML yourself.

For this example, I will have DataMgr copy the schema for the tables: users,groups,users2groups,teams.

First, instantiate DataMgr. For this example, the source database will be an MS SQL database with a datasource of "source".

```
<cfset DataMgr = CreateObject("component","DataMgr").init("source","MSSQL")>
```

Now, I need to load the tables that I want DataMgr to use. Normally, DataMgr will load a table upon first use. For this example, however, we aren't using DataMgr for anything but schema migration. So I need to specify which tables to use.

```
<cfset DataMgr.loadTable("users")>
<cfset DataMgr.loadTable("groups")>
<cfset DataMgr.loadTable("users2groups")>
<cfset DataMgr.loadTable("teams")>
```

In order to get the XML that DataMgr uses to define these tables, I just output the results of the getXml() method:

```
<cfoutput>#DataMgr.getXml()#</cfoutput>
```

I can save the results to a file (say "schema.xml") for later use (the XML can come from or work with any supported database).

To make sure that my product database has the correct structure, I first read the XML into a variable (say "dbschema") using cffile:

```
<cffile action="READ" file="#ExpandPath('schema.xml')#" variable="dbschema">
```

Now I instantiate DataMgr for the database on which I want to run the schema synchronization (a PostGreSQL database with a datasource of "target").

```
<cfset DataMgr = CreateObject("component","DataMgr").init("target","PostGreSQL")>
```

Then I pass the XML into DataMgr's loadXml() method:

```
<cfset DataMgr.loadXml(dbschema)>
```

DataMgr will create any missing tables and columns in the database. When you update the table structure, just run the code to create the XML again and replace your old XML file with the new one.

This works with any database that DataMgr supports (currently MS Access, MS SQL Server, MySQL, PostGreSQL - others should be easy to add).

DataMgr is open source and free for any use. You can **read more about it on my site** or **download it from RIAForge**.