Prototypes / QuerySim in DataMgr

Posted At : May 25, 2007 7:02 PM | Posted By : Steve
Related Categories: DataMgr

I got a call yesterday from a friend of mine who is just starting to use DataMgr. He asked me how to use the QuerySim feature of DataMgr which made me realize that I had never written a good explanation of how to use it.

First of all, it is important to understand that DataMgr can use **Database Introspection or Database Definition**. If you don't know when you are using, then you are probably using introspection.

DataMgr's simulated database (how it imitates **QuerySim**) requires database definition (after all, it doesn't have a real database to introspect). This means that you must pass in to DataMgr's loadXml() method an XML string to tell DataMgr the structure of your database.

If you have already designed your database structure in a database you won't have to hand-type the XML to define that structure. Just ask DataMgr for the XML.

To do that, first create an instance of DataMgr to point to your existing database (in this case a datasource of "testdb" pointing to a SQL Server database).

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

Now you can ask DataMgr what XML it would use to define this database structure:

```
<cfset DB_XML = DataMgrDB.getXml()>
```

Save that XML to a file for later use.

```
<cffile action="WRITE" file="#ExpandPath('db.xml')#" output="#DB_XML#">
```

This XML can now be passed in to the instance of DataMgr that is using the simulated database.

```
<cfset Application.DataMgr = CreateObject("component","DataMgr").init("","Sim")>

<cffile action="READ" file="#ExpandPath('db.xml')#" variable="DatabaseXML">

<cfset Application.DataMgr.loadXml(DatabaseXML,true,true)>
```

Now the Application.DataMgr instance (which is currently using the simulated databas, but which will use the real database when you want the application to start working) knows the structure of your database.

The preceding steps only have to be done once for your application. Now that you have done that, it is time to created a simulated query against a table.

If you have a table named "products", this code would produce a simulated query against that table with all of the fields (and correct data types) from the table.

```
<cfset qProducts = Application.DataMgr.getRecords("products")>
```

By default, the queries will have 50 rows of simulated data. If you want a different number, edit the "table" element for that table in your XML and add a "simrows" attribute with the number of rows of simulated data you want for that table.

Hopefully this provide a good introduction to simulated queries with DataMgr. If you have any questions, just let me know.

I also have an **Events demonstration** that show DataMgr at work. It randomly cycles between different databases (every hour or two). If you can catch it running on the simulated database, it makes a good example of the results of the simulated database (see the "Database" line near the top of the page).