

# Database Synchronization

Posted At : September 20, 2006 7:53 PM | Posted By : Steve

Related Categories: com.sebtools, DataMgr

In order to test some code for a recent project, I needed to have the data from the live site on my local computer. Of course, the live site is running MySQL which I don't have on my local computer.

By happenstance, I had just read a [blog entry](#) on transferring data from an Access to a MySQL database. It looked like a good solution (it uses DataMgr, it must be good!). Unfortunately, the solution listed only works if both databases have datasources on the same computer - not the case for me (though in retrospect I may have been able to set up a datasource on my local computer that pointed to the remote MySQL database).

In any event, I ended up building a tool to handle database synchronization. It will copy a database structure (and data unless you opt not to copy that) from one datasource to another.

This can be done one of two ways.

One:

1. Load the DataSynch.cfc and pass in the DataMgr for the source database.
2. run the getXml() method of DataSynch on the desired tables.
3. Take the resulting XML and pass it into the loadXML() of another DataMgr (this can be on another server running another database)

Two:

1. Load the DataSynch.cfc and pass in the DataMgr for the source database.
2. Call the addDataMgr() to pass in a DataMgr to which structure (and data) should be copied (this can be done multiple times, creating multiple receivers for structure and data)
3. Call the synchTables() method of DataSynch on the desired tables to synchronize the database structure and data with the datasources of each DataMgr pass in via addDataMgr()

Note that this approach won't update any existing data. The synchronization is one-way only (from the source datasource outward).

One other advantage to this approach is that DataSynch will do its best to figure out foreign keys. Instead of copying foreign keys by value, it will copy them by look-up (making sure that the value matches the primary key of the related record with the same data).

So, if you have a "categories" table with a catid field as an identity primary key field, the values created in the new table may not match the value in the old table. DataSynch will create the table so that it uses the new value for the catid field of your products table (assuming both are being synchronized) instead of the old value.

To borrow Michael's example. Here is how it could be implemented using DataSynch:

```
<!-- Create datamgr objects for each datasource -->
<cfset dmaccess = createobject("component","datamgr.DataMgr_Access").init("mydb_access")>
<cfset dmmysql = createobject("component","datamgr.DataMgr_MYSQL").init("mydb_mysql")>
<!-- Create the Synchronizer -->
<cfset synch = createobject("component","datamgr.DataSynch").init(dmaccess)>
<!-- Add the receiving datamgr -->
<cfset synch.addDataMgr(dmmysql)>
<!-- Synchronize the tables -->
<cfset synch.synchTables("contact,address,country,order,orderline,product,price")>
```

Because of the nature of DataMgr, any new tables or columns will automatically be created in receiving datasources.

**DataSynch** is free and open-source. It requires **DataMgr**, which can be downloaded from the same page.

