

DataMgr 2.0 Beta

Posted At : October 13, 2006 6:00 AM | Posted By : Steve

Related Categories: DataMgr

After using DataMgr 2 in production for a few weeks I think it is ready for beta. I am really excited about the new features. Even over using DataMgr 1.0, they have already saved me a lot of time.

You can [download DataMgr 2.0 Beta](#) and try it out. You can also download an upcoming version of [CodeCop](#) as an example implementation of DataMgr 2.0. Look in /sys/CodeCop.cfc for a thorough, working example of loadXML().

Here is a brief overview of the new features in DataMgr 2.0:

Relationships

DataMgr 2.0 has several built-in relationships (for lack of a better term). These relationships are not exactly the same as relationships in your database (and I would welcome a better term). Instead these relationships allow you to define some columns for DataMgr in a table that are not actual columns in the table in the database, but that DataMgr will use as columns. Instead of getting the data from the table, DataMgr will query another table for the data.

For example, you can have the category name for a product automatically included in the recordset for a product (using a "label" relationship).

You could also use a "list" relationship to get a list of values from another table (perhaps the primary key values from a many-to-many relationship - which could then be used in a checkbox).

Other types include: avg,count,max,min. The data from these relationships will be included in the recordset returned from getRecords(). You can also pass them in as data to getRecords() to filter the results. With the exception of the aggregate types (avg,count,max,min), you can also pass them into saveRecord() to update the data.

All relationships must be configured in loadXML().

Special Types

DataMgr can provide a number of different actions on special fields. All of which must be configured in loadXML().

Special Types:

- CreationDate
 - Only used with CF_SQL_DATE
 - DataMgr will automatically put the current date/time when inserting a record if no value is passed in for this field.
- DeletionMark
 - Only used with CF_SQL_BIT or CF_SQL_DATE
 - DataMgr will only retrieve records where this field is 0 (zero) for CF_SQL_BIT or NULL for either type.
 - If a table has a DeletionMark field, DataMgr will use it (setting it to true for CF_SQL_BIT or to the current date for CF_SQL_DATE) instead of deleting a record when calling deleteRecord().
- Sorter
 - Only used with CF_SQL_INTEGER
 - If no orderby argument is passed to getRecords(), this field will be the order by field.
 - When inserting a record, DataMgr will automatically assign this field to the next highest value (one higher than the highest value in the table) if no other value is passed in.

Username/Password

The `init()` method of `DataMgr` now accepts arguments for username and password. If both of these arguments are passed in, `DataMgr` will use them for the username and password attributes of `cfquery`.

Seed Data

Seed Data is not a new feature of `DataMgr 2.0`, but it is one that hasn't been covered and takes on increased importance in `DataMgr 2.0`. `DataMgr` can load data into a database table when it creates it or it can make sure that some data always exists in a table.

Simulated Database

With `DataMgr 2.0` comes a new database type. The database type "Sim" acts much like `QuerySim` in that it creates simulated data. Unlike `QuerySim`, however, it creates the data without you needing to type it. The simulated database will use seed data if any is given, otherwise it will generate random data.

The simulated database will also take advantage of relationships. It can include generated data and seeded data in one recordset if that is required by the relationship.

The "Sim" database is ideal for creating a prototype of an application. When you are ready to make your application functional, simply replace the call to `DataMgr_Sim` with a call to the `DataMgr` for your database.

The number of rows returned for generated data will be determined by the `rows` argument of the `init()` method or by a `simrows` attribute of the table tag in `loadXML()`.

Logging

You can log all changes to data done using `DataMgr`. The `startLogging()` method will cause `DataMgr` to store a record for every insert, update or delete. By default this will be done in the "datamgrLogs" table.

You can also use the `stopLogging()` method to stop logging in `DataMgr`.

getRecords()

As a small change, `getRecord()` has new arguments. The `orderBy` arguments will order the records (superceding any "Sorter" field). The `maxrows` argument will limit the number of records returned. The `fieldlist` argument will limit the fields returned by `getRecords`. This can include relationship fields.

Synchronizer

Although technically not a new feature, the `DataSynch.cfc` is also available to synchronize databases.

/database_synchronization.htm

More details on these features - and how to use them - can be found in the word doc that is included with the [DataMgr 2.0 Beta zip file](#) . Additionally, I plan to do more blog entries and/or presentations about individual features.

While the new version has been working great for me, I expect that you may find improvements that can be made to the API or implementation. If you do, [let me know](#) .

[Download DataMgr 2.0 Beta.](#)

[Find out more about DataMgr.](#)