Database Introspection or Database Definition

Posted At : May 25, 2007 4:59 PM | Posted By : Steve
Related Categories: Frameworks, DataMgr

In most of our interactions between code and database, one dictates the other. If we are working with a legacy database, then the structure of the database will dictate how we write our code. If we are proceeding from a visual prototype, then our database follows from our design.

Similarly, tools that we use for database interaction take one of these mindsets. Neither are necessarily right or wrong, but which we choose is largely based on our situation.

The popular ORM framework, **Transfer** , requires that it be given a definition of the structure of the tables that it uses. If you don't pass in XML to define your table, then it won't be able to work.

Another popular ORM framework, **Reactor**, is largely based on database introspection (but does require some information about relationships).

It looks like **objectBreeze** (another ORM) uses database introspection as well.

As for my own creation **DataMgr** (not an ORM), it gives you a choice. If you don't tell it anything about a table, it will introspect the database to find out about it. If you do, it will use the information that you tell it (and optionally create missing tables and columns to match).

Ultimately, knowing your own style and the options available should help you choose. If you are using OO then an ORM makes sense. Then you need to decide if you want to define the database or use introspection.

Incidentally, I am basing my assessments of each ORM based on what I could find y brief perusal of their documentation. If I have any inaccuracies, let me know and I will correct them.