

Backward Compatibility - Class Rename - Meta Attribute

Whenever you put an object in a DataItem, NHibernate figures out how to save this 'Value' entity using what is called an Any mapping. As part of this Any mapping mechanism, NHibernate stores the (CLR, .NET) type of the object in a column in the database as fully qualified string. Also sometimes we map types directly, for example ValueType of a dataitem.

This is all fine, until you rename a class..

When NHibernate tries to convert the string back into a .NET type, it will fail and crash. Legacy mappings won't help you there.

To solve this issue, we have added a custom HBM (meta)attribute in DeltaShell: oldClassName. The value of this attribute is used as an alias when resolving types. This happens automatically for the any mapping of dataitem, but if you have mapped a Type column/property directly, make sure to load it using UserType in your legacy mapping.

Example:

Your old class 'Branch' has been renamed to 'Channel'. Now in your legacy mapping you do the following:

Legacy Mapping

```
<class name="Channel" table="branch" lazy="false">

  <meta attribute="oldClassName">Domain.Branch, Domain</meta>

  <id name="Id" column="Id" type="Int64" unsaved-value="0"><generator class="native" /></id>
  ...
</class>
```

The value of the 'oldClassName' attribute is the string as it appears in the old database, typically 'namespace.class, assembly'.



To summarize: if you rename a persistent class, and there is any chance the object has ever been used as DataItem value, parameter, variable, filter, or had its type mapped directly for any other reason: make sure to add a meta attribute with the old class alias.