How can I manage Ensemble Record Maps?

04/26/2024 11:57:51

FAQ Article Print

Category:	Products::Deltanji (formerly VC/m)	**Votes:**	0
State:	public (all)	**Result:**	0.0 %
		Last update:	16:24:56 - 09/27/2018

Symptom (public)

You are trying to manage your Ensemble Record Maps in Deltanji so you can version them and transfer them between environments. You need to understand the code entities associated record maps, and how to manage them.

Problem (public)

Record Maps are implemented by a set of two or more classes. The primary class (known as the RecordMap class) extends EnsLib.RecordMap.RecordMap and comprises an XData section and a couple of get and put methods. The second class (known as the Record class) is typically a generated one, and by default is named by appending a .Record suffix onto the name of the primary class. The record class is used to persist rows of record data.

For example:

- User.Customer extends EnsLib.RecordMap.RecordMap and its XData section contains /Record/@targetclassname="User.Customer.Record"
- User.Customer.Record extends %Persistent, Ens.Request, EnsLib.RecordMap.Base and is normally generated from the main class, User.Customer. It has a parameter RECORDMAPGENERATED whose value is 1 if this is the case.

Some record maps generate more than one class. The extra generated classes are typically serial classes used by the record class. For example, in the ENSDEMO namespace:

- Demo.RecordMap.Map.FixedWidth generates Demo.RecordMap.Map.FixedWidth.Record and Demo.RecordMap.Map.FixedWidth.Record.HomeAddress
- Demo.RecordMap.Map.Delimited generates Demo.RecordMap.Map.Delimited.Record, Demo.RecordMap.Map.Delimited.Record.HomeAddress and Demo.RecordMap.Map.Delimited.Record.WorkAddress

Some record maps are configured so that their records get assembled into batches. In this case its RecordMap class specifies a generated batch class which has a one-to-many relationship with the Record class. In ENSDEMO the Demo.RecordMap.Map.Delimited recordmap class specifies Demo.RecordMap.Map.Delimited.Batch as its batch class, and that class has a one-to-many relationship with the Demo.RecordMap.Map.Delimited.Record record class.

If it becomes necessary to make manual modifications to a generated class, the programmer should set RECORDMAPGENERATED=0 to stop Portal's Generate button overwriting the manual changes.

Solution (public)

The following are two ways to approach the management of Record Maps with Deltanji:

- Register the main class, and manually regenerate the other classes after any transfer.
- Register the main class plus the generated classes as multiple components of the same Object.

George James Software recommends the second way, which makes use of Deltanji's ability for an object to contain more than one component. A typical Deltanji object contains a single component, but Deltanji allows you to associate more than one component with a single object. Doing this binds these components together such that if any one of them is transferred all the other components will automatically be transferred at the same time.

To add a second component to an existing object that already contains a component:

- Make sure the existing object is checked out.
- Navigate to the second component using the Component view of the location. You should see that the component is unregistered.
- Copy the component to your Work List, either by selecting it and typing Ctrl+C or by right clicking and selecting Copy to Work List.
- Navigate to the object using the Object view and select it.
- Right click and select Paste from Work List. You will be asked to confirm that you want to add the component to the object.
- If you inspect the object you will see that it now lists two items in its component list.