Mapping caIntegrator2 object model to data model

Step 1) Make sure the object model and data model have both been updated and follow the caCORE modeling conventions as detailed in the caCORE SDK Developer’s Guide

* Tables are all upper case named as well as columns
* Classes all use camel case and do not use key words
* Attributes in classes all use java primitive types or other classes in the model
* Associations all have multiplicity set
* All associations are properly named including using Collection on the end of any multiplicity associations

Step 2) Export the entire Logical View from EA into the caintegrator2.xmi file using the default export options from EA

Step 3) Launch caAdapter 4.0 and choose File->Open->Object Model to Data Model Mapping File and select the caintegrator2.xmi file

Step 4) Map any new attributes or associations that were added to the model

* Existing mappings will be shown
* Selecting the Validate button from the toolbar will validate the mappings to ensure that all required mappings have been done
* INFO messages can be ignored but not ERROR messages
* For more details on how to map using caAdapter see the caAdapter User’s Guide

Step 5) Save the mapping using the Save button on the toolbar

Step 6) Import the caintegrator2.xmi file into EA

* Open the client EAP and delete the Data Model and Logical Model packages
* Import Model from XMI

Step 7) Save the client EAP which now includes the new mappings which will be exported with future changes

Step 8) Export the model to caintegrator2.xmi once more

* For some reason the modified version by caAdapter does not run through the SDK but exporting it again from EA seems to fix it
* Run the build to ensure everything worked

Step 9) Check the client EAP and the new XMI in