Article
Robert Hurst · Feb 3, 2021 3m read

Ensemble Operation: Calling a SQL Stored Procedure

We use the Caché **JDBC Gateway** to Oracle and SQL servers to directly invoke their stored procedures from Ensemble. Getting quick, inline data results back are typically handled within the **Functions.Library** class as a function to wrap the query and format the return appropriately.

But what about more elaborate stored procedures made for posting data without log-jamming a Router? The operations to construct those data pipelines can get tedious, especially when changes are needed. What follows is a code generator to make a new **Ensemble Message** class from a SQL Stored Procedure that is compatible to send off to a Business Operation. Its package source listing is attached.

Working example:

Step 1) Import the SQL Stored Procedure Call via SMP Explorer: **SQL Link Wizard**

Step 2) Create your **new project class** that points to the SQL proxy class name created by the Wizard:

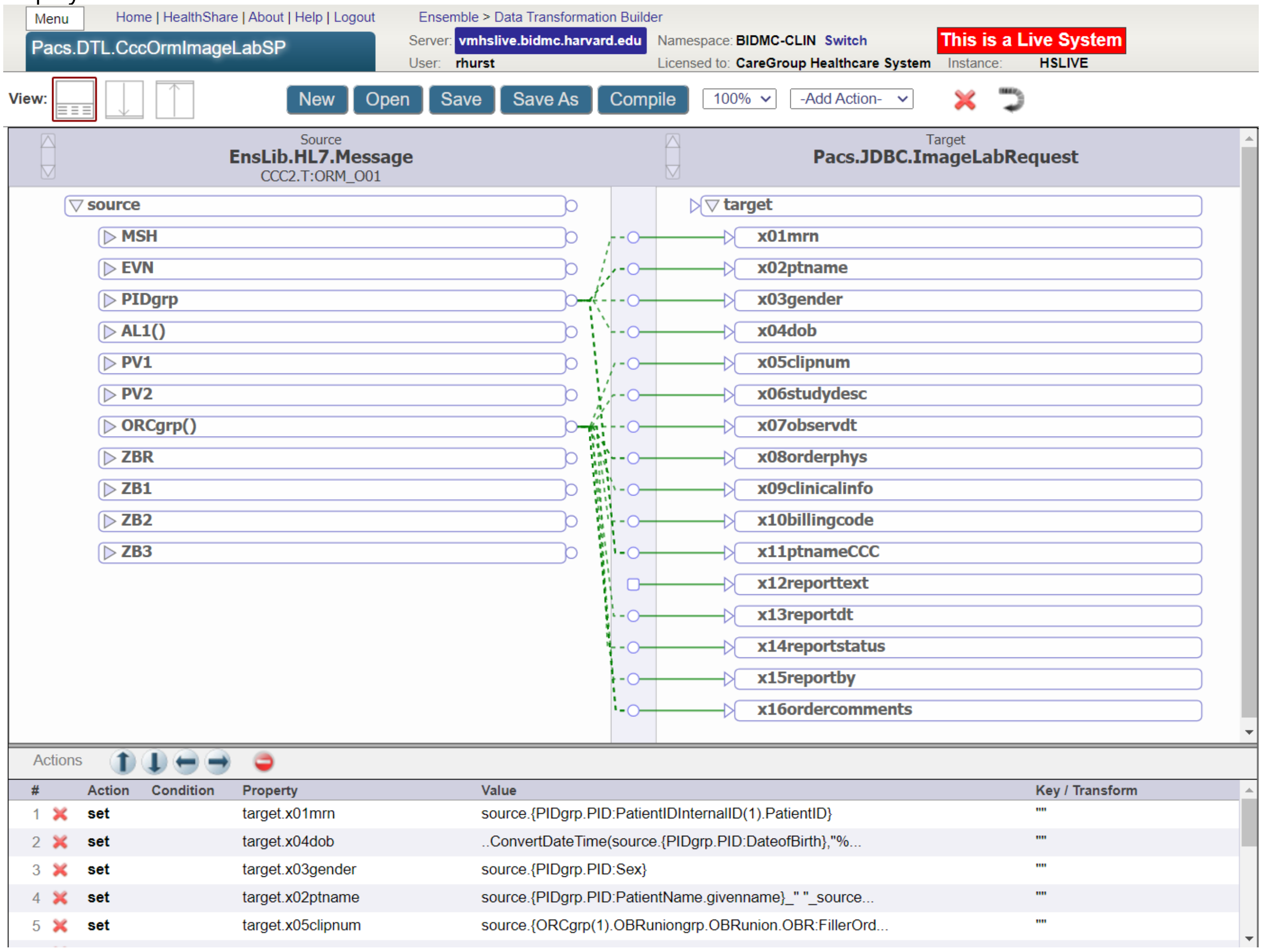
```
Class Pacs.JDBC.MakeImageLab Extends Common.JDBC.MakeRequestClass
{
  // supply the name of the JDBC proxy class created by SQL Link Procedure Wizard.
  Parameter LINK = "dbo.Imagehl7add1";
  // supply the name of the generated message class for DTL
  Parameter MAKE = "Pacs.JDBC.ImageLabRequest";
}
```

Step 3) Upon **Save & Compile** of Step #2, **a new Message class generates** using the name supplied to **MAKE**

```
/// DO NOT TOUCH!
/// auto-generated by Common.JDBC.MakeRequestClass via Pacs.JDBC.MakeImageLab
/// Authored by Robert Hurst
Class Pacs.JDBC.ImageLabRequest Extends
Common.JDBC.StoredProcedureRequest [ GeneratedBy
= Pacs.JDBC.MakeImageLab.CLS, Not ProcedureBlock ]
{
  Property x01mrn As %String(MAXLEN = 20);
  Property x02ptname As %String(MAXLEN = 50);
  Property x03gender As %String(MAXLEN = 1);
  Property x04dob As %TimeStamp;
  Property x05clipnum As %String(MAXLEN = 75);
  Property x06studydesc As %String(MAXLEN = 200);
  Property x07observdt As %TimeStamp;
  Property x08orderphys As %String(MAXLEN = 80);
  Property x09clinicalinfo As %String(MAXLEN = 300);
  Property x10billingcode As %String(MAXLEN = 20);
  Property x11ptnameCCC As %String(MAXLEN = 50);
  Property x12reporttext As %Stream.GlobalCharacter;
  Property x13reportdt As %TimeStamp;
  Property x14reportstatus As %String(MAXLEN = 12);
  Property x15reportby As %String(MAXLEN = 500);
  Property x16ordercomments As %String(MAXLEN = 2500);

  /// supply whether message class is a group of messages, or not
  Parameter GROUP As INTEGER = 0; ///
the name of the JDBC proxy class created by SQL Link Procedure Wizard
  Parameter LINK = "dbo.Imagehl7add1";
 }
```

Note the prefix used in front of each parameter name. That is to enforce its ordinal position when it goes to display within the **DTL** Builder:



#	Action	Condition	Property	Value	Key / Transform
1	set		target.x01mrn	source.{PIDgrp.PID:PatientIDInternalID(1).PatientID}	""
2	set		target.x04dob	..ConvertDateTime(source.{PIDgrp.PID:DateofBirth},"%...	""
3	set		target.x03gender	source.{PIDgrp.PID:Sex}	""
4	set		target.x02ptname	source.{PIDgrp.PID:PatientName.givenname}_" "_source...	""
5	set		target.x05clipnum	source.{ORCgrp(1).OBRuniongrp.OBRunion.OBR:FillerOrd...	""

Step 4) Create your target Business Operation using the **JDBC.StoredProcedureOperation** class, supplying the inbound message name into its *Additional Settings* **RequestClassname** field presented. In this working example, the target message classname out of the DTL: **Pacs.JDBC.ImageLabRequest**

Rinse-lather-repeat with other SQL stored procedures. Onward!

#Business Operation #Ensemble

Source URL:https://community.intersystems.com/post/ensemble-operation-calling-sql-stored-procedure