How to Use SAP Enterprise Services in Visual Composer for Composition Environment Applications

Applies to:

Visual Composer for enhancement package 1 for SAP NetWeaver Composition Environment 7.1.

Summary

This article provides an introduction to clustered data services and describes how to model a Visual Composer application using the Services Registry.

Author: Netanel Slomianski

Company: SAP

Created on: 16 December 2008

Author Bio

Netanel Slomianski is a Visual Composer solution expert working for the Visual Composer Solution Office group.

Table of Contents

Purpose

Visual Composer provides the ability to model applications using various data sources such as BAPIs, Web services, and databases. It also enables you to access the Services Registry to locate and use data sources in your models. In this How-To guide, we will model an application that displays sales order data, using the Services Registry.

Find out how to configure Visual Composer to consume enterprise services from the service registry in this how-to guide.

Step 1: Creating a New Model

From the main menu, choose File→New. This opens a dialog box in which you can create a new Visual Composer component. Choose Composite View (the standard Visual Composer application), enter a name for the component, choose the development component in which to create the model, and click OK.

The model is created and an empty workspace is displayed, as shown in Figure 1.

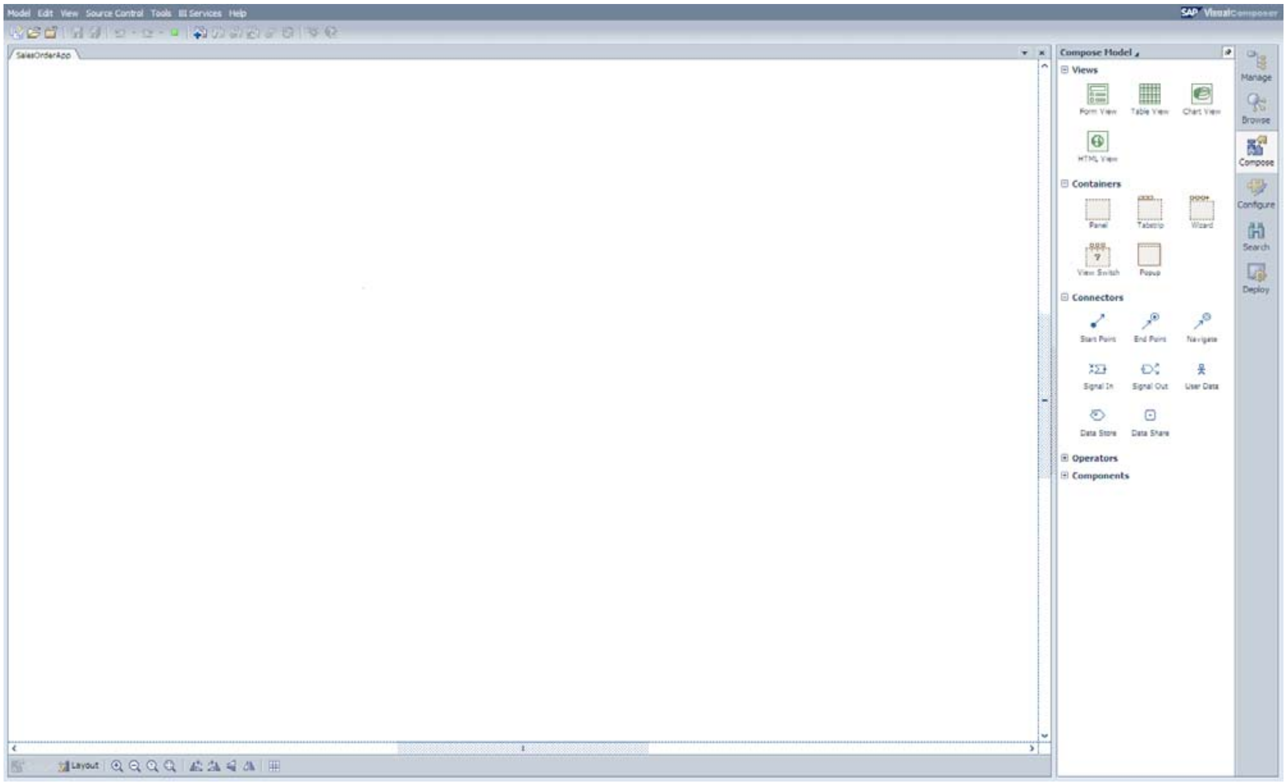
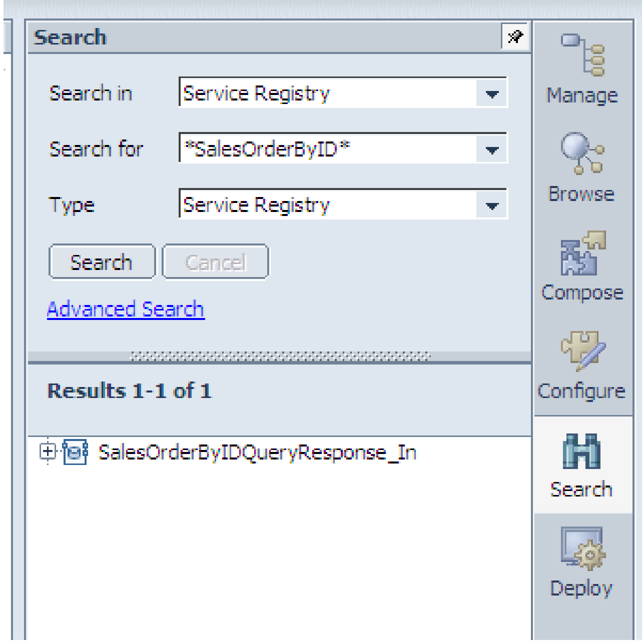


Figure 1

Step 2: Importing the Service from the Services Registry

In this step, we will search for the required service and import it to the model.

Select the Search task panel and choose Services Registry in the *Search In* field. In the *Search for* field, enter the name of the service you want to search for: in this case it is *SalesOrderByID* (the search also accepts wildcards). Click *Search*. Figure 2 depicts the Search pane as it should appear after clicking the Search button.



Search

Search in: Service Registry

Search for: *SalesOrderByID*

Type: Service Registry

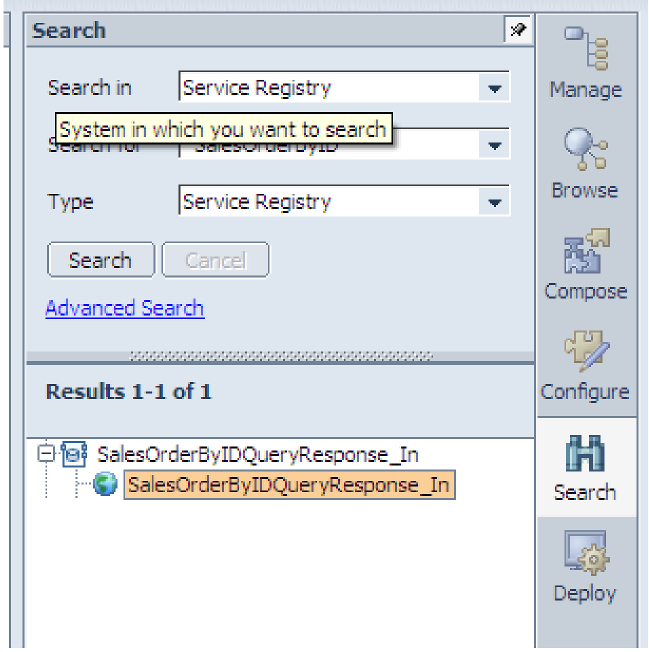
Results 1-1 of 1

SalesOrderByIDQueryResponse_In

Figure 2

Click the plus sign to the left of the service to view the actual operations it implements (end-points).

In this case, there is only one, as can be seen in Figure 3.



Search

Search in: Service Registry

System in which you want to search

Type: Service Registry

Results 1-1 of 1

SalesOrderByIDQueryResponse_In

SalesOrderByIDQueryResponse_In

Figure 3

Drag and drop the operation onto the Design board. The Define Ports dialog box opens. This dialog box enables you to select the ports and fields of the service that you want to use. Select both the SalesOrderByIDQuery_sync in port and the SalesOrderByIDResponse_sync out port. All of the child nodes and fields are selected automatically.

Step 3: Displaying the Data in UI Elements

In this step, we will create UI elements (a form and two tables), which display the data returned from the service.

To create the form, drag out from the service's input port and choose Form View (Figure 4).

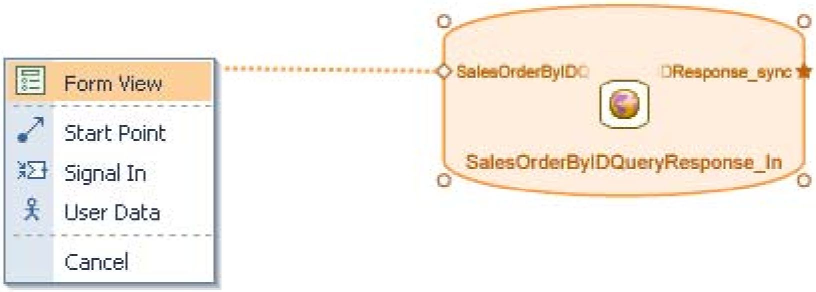
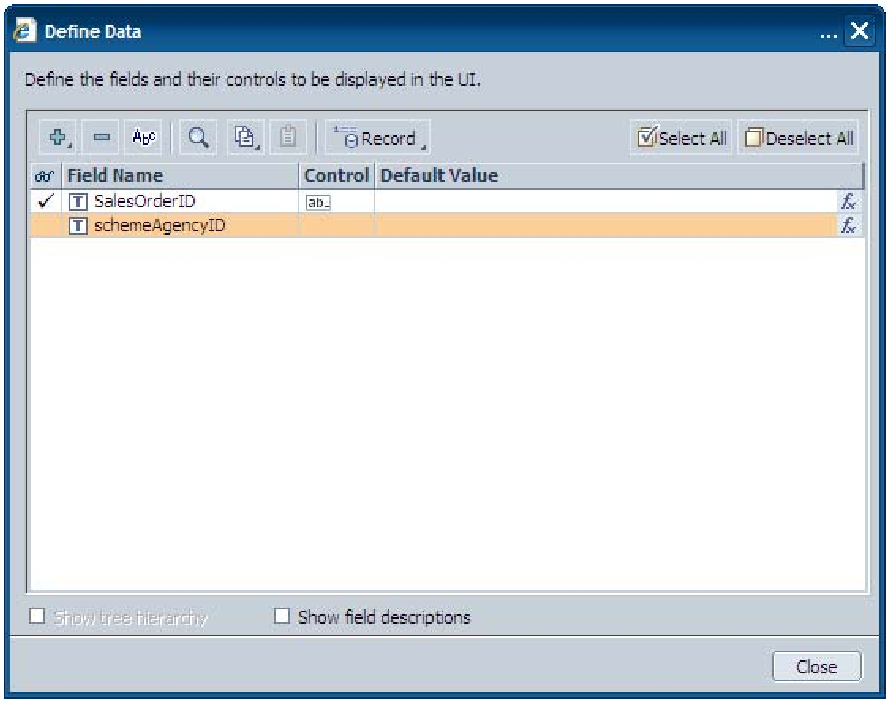


Figure 4

A form is created. Right-click the form and choose *Define Data.* In the Define Data dialog box, deselect the control for the schemeAgencyID field, since it is not required as input to the service (Figure 5) and click *Close*.



Define the fields and their controls to be displayed in the UI.

	Field Name	Control	Default Value
✓	SalesOrderID	ab	
	schemeAgencyID		

□ Show tree hierarchy

□ Show field descriptions

Figure 5

The next step is to create the tables that will show the service's output data. Drag a table out of the service's out port (Figure 6).

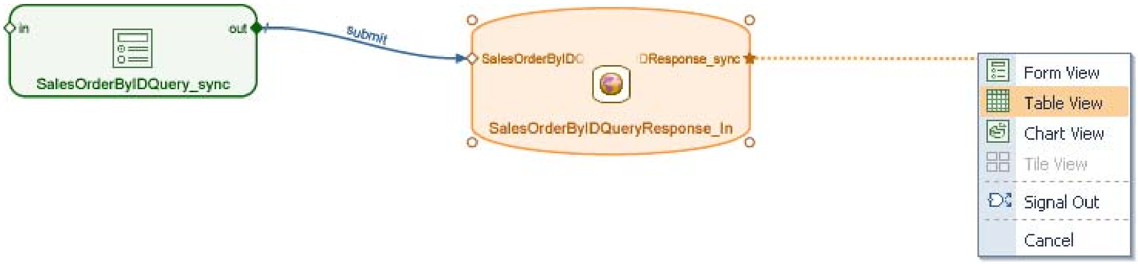


Figure 6

Since there are a lot of fields in this port, the Define Data dialog box opens automatically, so that you can select the fields that you want to display in the table.

Use the following table to determine which fields to select:

Node Name	Fields to Select
SalesOrderByIDResponse_sync	Sales_order→Date
	SalesOrder→GoodsRecipientParty→FormattedName→FormattedName
	SalesOrder→DateTerms→RequestDate
	SalesOrder→TotalValues→GrossWeightMeasure→GrossWeightMeasure
	SalesOrder→TotalValues→NetAmount→NetAmount
	SalesOrder→InvoiceTerms→InvoicingDate

Click Close. The table is created in your model (Figure 7).

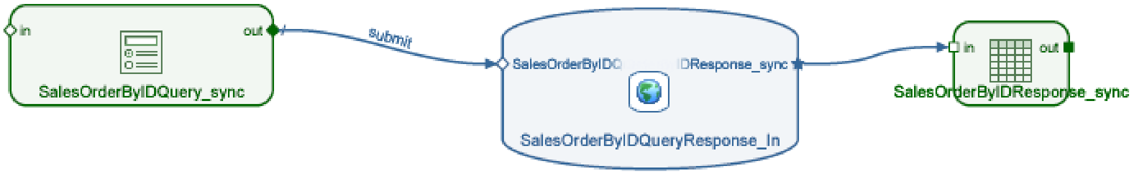


Figure 7

This table displays the customer's sales order details. To display the sales order items, we create another table that shows the items that are included inside the sales order.

To do this, drag a table from the ***SalesOrderByIDResponse_sync*** table and use the following table to determine which fields to select:

Node Name	Selected Fields
SalesOrderByIDResponse_sync ↳ SalesOrder_Item	Description.Description
	Product.SellerID
	BuyerParty.FormattedName.FormattedName
	GoodsRecipientParty.FormattedName.FormattedName
	PayerParty.FormattedName.FormattedName
	DeliveryTerms.PriorityValue
	DeliveryTerms.Incoterms.TransferLocationName
	InvoiceTerms.InvoiceDate
	TotalValues.RequestedQuantity.RequestedQuantity
	TotalValues.ConfirmedQuantity.ConfirmedQuantity
	TotalValues.NetPrice.Amount.Amount

Click OK. The table is created (Figure 8).

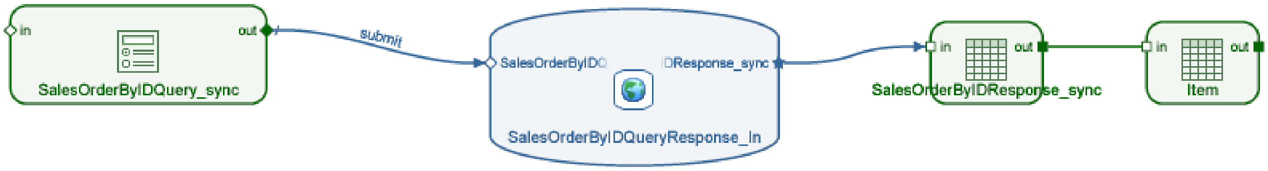


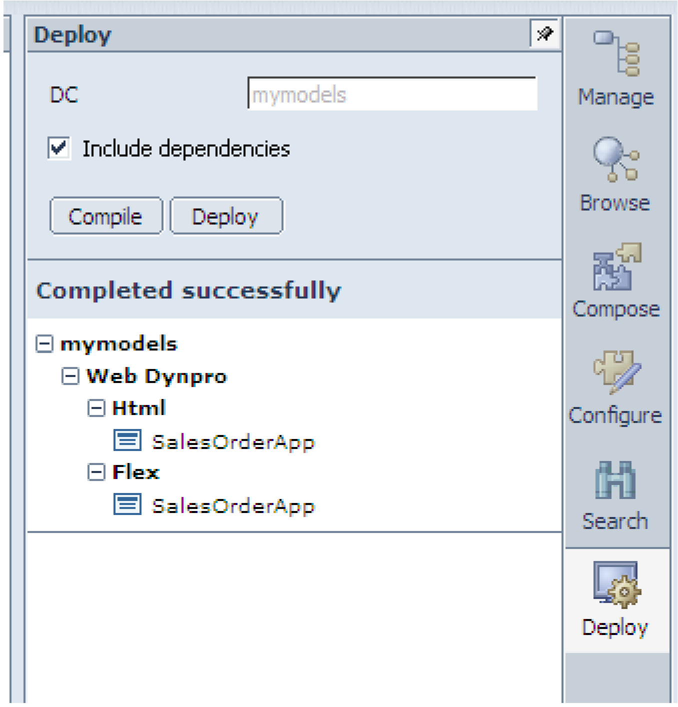
Figure 8

You can switch to the Layout board by right-clicking any of the tables or form and choosing Layout Board. Here you can change how the tables and form are displayed and also the titles of the form, the tables and the table's fields, by selecting the element and changing its properties in the Configure task panel.

Step 4: Deploying and Running Your Model

In this step, we will deploy the model on the server. Select the *Deploy* task panel and click the *Deploy* button.

After the deployment is complete, two links are displayed that open the deployed model. The first one opens the application in Web Dynpro HTML runtime and the second one opens it in the Web Dynpro Flex runtime. Choose one of the links to run the application (Figure 9).



Deploy

DC mymodels

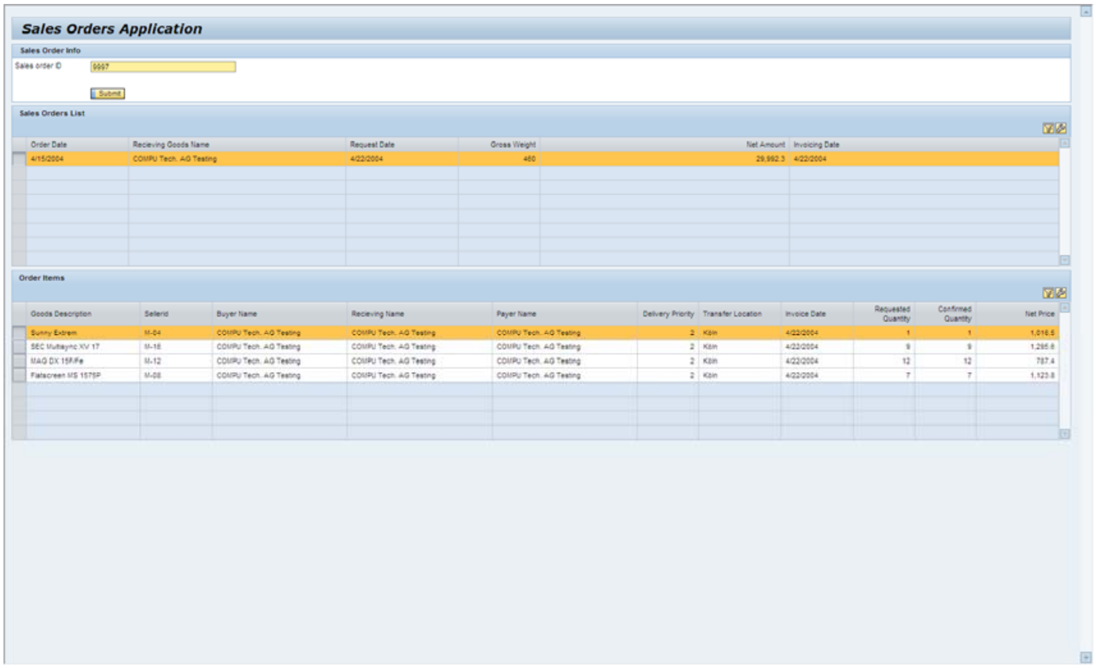
☑ Include dependencies

Completed successfully

- mymodels
  - Web Dynpro
    - Html
      - SalesOrderApp
    - Flex
      - SalesOrderApp

Figure 9

That's it. Your first Visual Composer application with a Services Registry service is ready and running! (Figure 10)



Sales order ID: 9997

Sales Orders List

Order Date	Recieving Goods Name	Request Date	Gross Weight	Net Amount	Invoicing Date
4/15/2004	COMPU Tech. AG Testing	4/22/2004	460	29,992.3	4/22/2004

Order Items

Goods Description	Sellerid	Buyer Name	Recieving Name	Payer Name	Delivery Priority	Transfer Location	Invoice Date	Requested Quantity	Confirmed Quantity	Net Price
Sunny Extrem	M-04	COMPU Tech. AG Testing	COMPU Tech. AG Testing	COMPU Tech. AG Testing	2	Köln	4/22/2004	1	1	1,016.5
SEC Multisync XV 17	M-18	COMPU Tech. AG Testing	COMPU Tech. AG Testing	COMPU Tech. AG Testing	2	Köln	4/22/2004	9	9	1,295.8
MAG DX 15F/Fe	M-12	COMPU Tech. AG Testing	COMPU Tech. AG Testing	COMPU Tech. AG Testing	2	Köln	4/22/2004	12	12	787.4
Flatscreen MS 1575P	M-08	COMPU Tech. AG Testing	COMPU Tech. AG Testing	COMPU Tech. AG Testing	2	Köln	4/22/2004	7	7	1,123.8

Figure 10

Related Content

How to Configure Visual Composer to Consume Enterprise Services from the Service Registry

Visual Composer for SAP NetWeaver CE 7.1

ES Workplace Handbook

Services Registry for ESWorkplace

Copyright

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided “as is” without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Any software coding and/or code lines/strings (“Code”) included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.