# Creation of VC Model Using Exchange Rate Web Service

SAP DEVELOPER NETWORK

## Applies To:

SAP Netweaver 2004s
Visual Composer 7.0

## Summary

The purpose of this document is to show how to create a VC model using the exchange rate web service call and updating the result of the call in SAP.

**By**: Velu Pillai
**Company**: Intel Technology India Pvt. Ltd
**Date**: 03 August 2006

## Table of Contents

## Introduction

This technical document is based on my earlier weblog in SDN (https://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/docs/library/uuid/6b5de790-0201-0010-3cb9-f03e8c4aeb12) using the previous version of Visual Composer. In that web log, I tried the same scenario using the function module wrapper to call the web service to convert the exchange rate (http://www.webservicex.net/CurrencyConvertor.asmx ). Now with the general availability of Visual Composer NW04s, the web service call is natively supported in the tool. So I thought of creating the same model using Visual Composer NW04s.

## Create Model

Now we will move our focus to, the creation of VC model with updating the exchange rates real time in SAP using the currency converter web service call. You need to first login to the visual composer server by specifying the URL below. For example, the URL could be,

http://<machine name>:<portnumber>/VC/default.jsp

After logon to the client, create a model named "CURRENCY_CONVERT_MODEL" and then embed an empty iView into your model by dragging the "iView" object from the component task panel and name it as "Exchange Conversion".

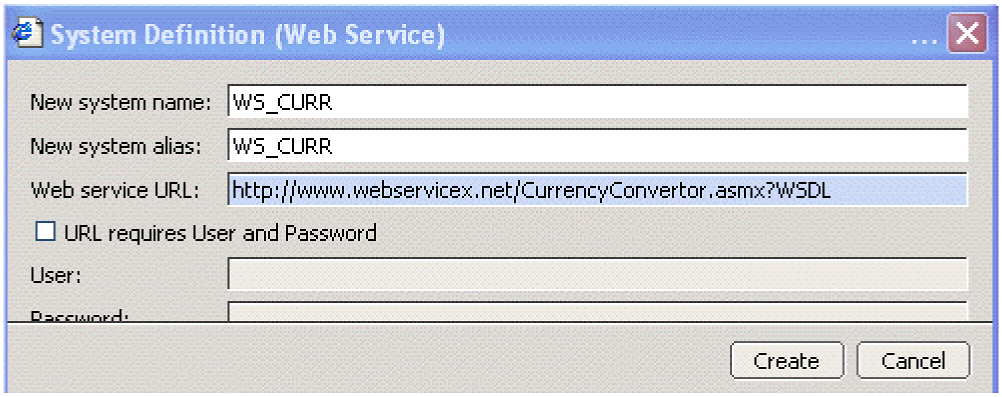
In our scenario, we will be modeling an application which will make first a call to an external web service for converting the exchange rate and then we will update the exchange rates from the result of the web service call in SAP (using the BAPI function call "BAPI_EXCHANGERATE_CREATE ") and then the updated values will be displayed in table (using the BAPI call "BAPI_EXCHRATE_GETCURRENTRATES ").

## Data Services

Next step is to create the data services for the following functionalities.

To call the web service for exchange rate conversion.
Update the exchange rate from the result of the web service call in SAP.
Display the updated exchanges from SAP in a table view.

First we will create the data service for calling the web service, by clicking on the menu option "Tools" and then "Define web service system". Fill out the details for the web service system by specifying the system name, alias and WSDL URL of the web service call like below:



New system name: WS_CURR

New system alias: WS_CURR

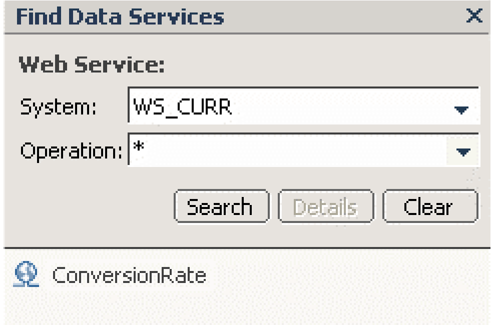
Web service URL: http://www.webservicex.net/CurrencyConvertor.asmx?WSDL

☐ URL requires User and Password

User:

Password:

This will make a connection to the portal and creates the web service system in the portal system landscape. Next we will create a data service for the web service call, by clicking on the “Find Data Services” in task-panel toolbar. This will make a connection to the portal system landscape and brings back the list of system defined in the portal. Select the system “WS_CURR” from the drop down to create the data services like below (In operation, type in “*”):



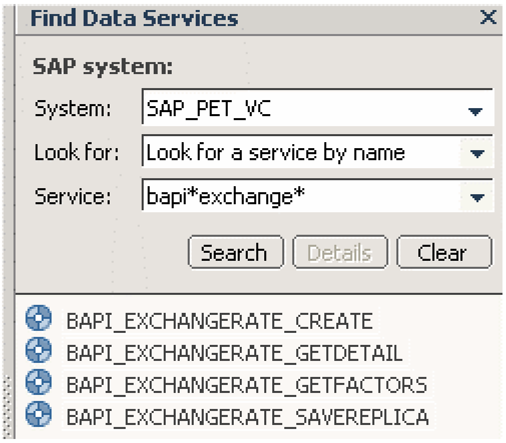
System: WS_CURR

Operation: *

ConversionRate

Select the operation service “ConversionRate” and drag and drop into your model and then create an input form by just dragging the input connector from the data service. Next step is to create output form by just dragging the output connector from the data service.

Next step is to update the exchange rate from the result of the web service call in SAP, by calling the data service “BAPI_EXCHANGERATE_CREATE.” For this we need to select the SAP backend system from the drop down to create the data services like below (In service, type in “BAPI*EXCHANGE*”):



SAP system:

System: SAP_PET_VC

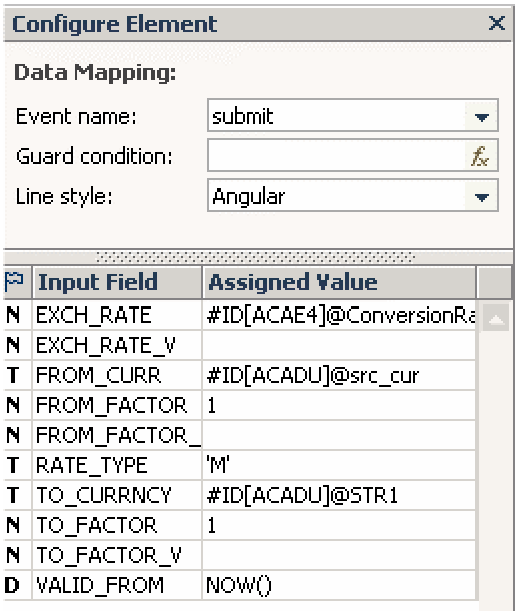
Look for: Look for a service by name

Service: bapi*exchange*

- BAPI_EXCHANGERATE_CREATE
- BAPI_EXCHANGERATE_GETDETAIL
- BAPI_EXCHANGERATE_GETFACTORS
- BAPI_EXCHANGERATE_SAVEREPLICA

Select the data service “BAPI_EXCHANGERATE_CREATE” and drag and drop into your model and then create a link by just dragging the input connector from the data service and connect it into the output of the web service form. In order to update the exchange rate in SAP, we need to call the transaction commit after the first BAPI call. This can be done by calling the data service “BAPI_TRANSACTION_COMMIT”.

Here is the data mapping between the operation service “ConversionRate” and the data service “BAPI_EXCHANGERATE_CREATE”.



Data Mapping:

Event name: submit

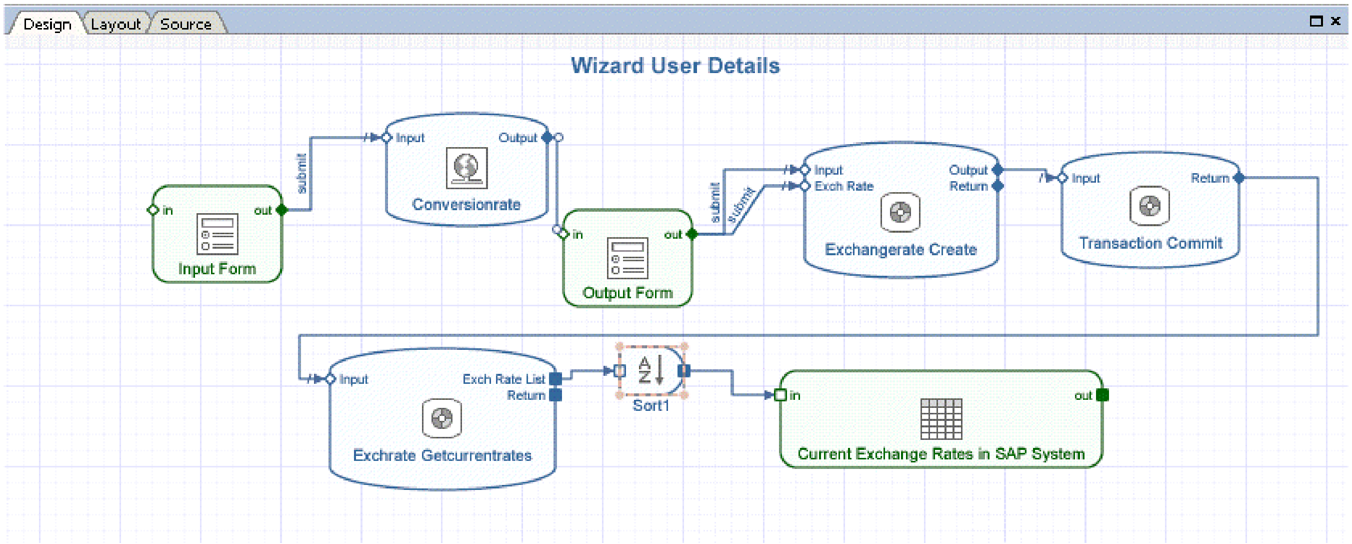
Guard condition:

Line style: Angular

| | Input Field | Assigned Value |
|---|---|---|
| N | EXCH_RATE | #ID[ACAE4]@ConversionRa |
| N | EXCH_RATE_V | |
| T | FROM_CURR | #ID[ACADU]@src_cur |
| N | FROM_FACTOR | 1 |
| N | FROM_FACTOR_ | |
| T | RATE_TYPE | 'M' |
| T | TO_CURRNCY | #ID[ACADU]@STR1 |
| N | TO_FACTOR | 1 |
| N | TO_FACTOR_V | |
| D | VALID_FROM | NOW() |

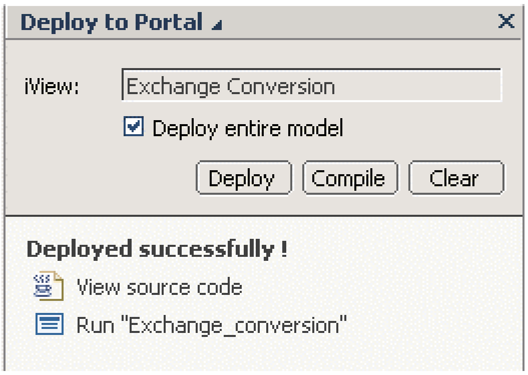
Repeat the above steps for the data service “BAPI_TRANSACTION_COMMIT” and connect the output of the exchange rate create data service into the input of the BAPI transaction commit data service.

Next step is displaying the updated exchange rates from SAP in a table view.  For this you need to bring the data service "BAPI_EXCHRATE_GETCURRENTRATES" from your back end system to the model and connect the output of the BAPI transaction commit data service into the input of the data service "Getcurrentrates".  Connect the output from the data service to create the output table view for displaying the updated exchange rates from SAP. In order to the show the most recent updated values in the table view, we may need to call the operator "Sort" for sorting the exchange rate results by 'VALID_FROM' date. After performing all the steps, your model will look below:



## Deploy & Test your application

Please make sure that you have the necessary access permissions to deploy the applications into your PCD (Portal Content Directory) under the folder "Visual Composer" or any other folder configured in the deployment folder option. Click on the "Deploy" button at the main toolbar or the "Deploy" button in task-panel toolbar. While deploying the model, system will compile the model first and then deploy it into the portal content directory. You can run the application, after the successful deployment by click on the "Run" button in the task panel like below:


Deploy to Portal

iView: Exchange Conversion

☑ Deploy entire model

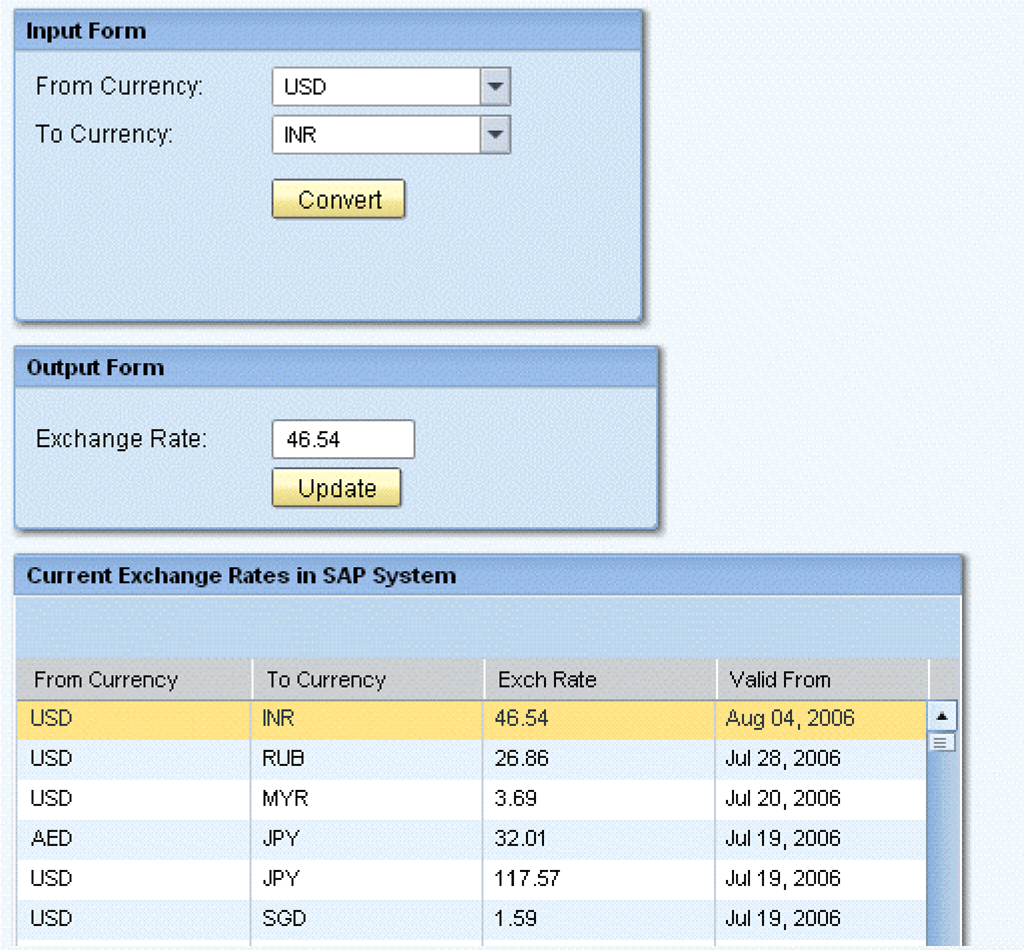
Deploy | Compile | Clear

Deployed successfully !

View source code

Run "Exchange_conversion"

As the result of run, you can see the application in portal like below:



**Input Form**

| | |
|---|---|
| From Currency: | USD |
| To Currency: | INR |

**Output Form**

| | |
|---|---|
| Exchange Rate: | 46.54 |

**Current Exchange Rates in SAP System**

| From Currency | To Currency | Exch Rate | Valid From |
|---|---|---|---|
| USD | INR | 46.54 | Aug 04, 2006 |
| USD | RUB | 26.86 | Jul 28, 2006 |
| USD | MYR | 3.69 | Jul 20, 2006 |
| AED | JPY | 32.01 | Jul 19, 2006 |
| USD | JPY | 117.57 | Jul 19, 2006 |
| USD | SGD | 1.59 | Jul 19, 2006 |

## Pre-requisite

In order to log on to and work with Visual Composer Storyboard, be sure that your Visual Composer client machine meets the following software requirements:

- Microsoft Internet Explorer 6.0 SP1 or higher.
- Microsoft XML Parser 4.0.
- Adobe SVG Viewer 3.0 or higher (downloadable from http://www.adobe.com/svg).
- Adobe (formerly Macromedia) Flash plug-in, for viewing models deployed to the Flex environment.

You need to have the necessary access permissions to create the system in the system landscape directory of your SAP Netweaver portal.

## Author Bio

Velu Pillai is working in Intel Technology India Pvt. Ltd as a Senior Application developer. He is currently working on the Enhancement/Switch framework, Visual Composer and Web dynpro for ABAP as part of the Netweaver 2004s exploration. This is his second article on “Visual Composer”.  He has already published an article on “Visual Composer – Creating exchange conversion model using an external web service” in SDN.  He is also an active member of ASUG and he recently presented a session on “Understanding RFID Data Management: SAP Auto-ID Infrastructure at Intel” in SAPPHIRE 2006.

## Disclaimer & Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.