

Using Signal Name as Event in Visual Composer Models



Applies to:

Visual Composer for SAP NetWeaver 7.0, SP Stack 14 Patch and above

Summary

The Signal Name as Event feature, enabled by the new Flash environment, exposes an internal event within a nested iView at the port of the Signal Out connector. The containing iView can listen for each event and use the signals to transfer parameters between nested iViews within the same iView.

Author: Judy Kestecher

Company: SAP Labs Israel

Created on: March 3, 2008

Author Bio



Judy Kestecher is an information developer working in the NetWeaver information development team at SAP Labs Israel. She is responsible for coordinating the documentation of Visual Composer for the different NetWeaver platforms.

Table of Contents

Overview	3
The Use Case	3
Steps in this Scenario	3
The Model – Overall Description	5
Implementations of Signal Name as Event:.....	6
In summary,	7
Copyright.....	8

Overview

The recently-released Visual Composer patch for SAP NetWeaver 7.0 SP Stack 14 includes a new feature that enables the modeler to trigger events between nested iViews via the out port of the nested iView. The means of achieving this is based on using *Signal Out* elements within the nested iViews. The result is that the iView containing the nested iView triggers an action assigned the *Signal Out* name each time the *Signal Out* is updated, and thereby “listens” for the action.

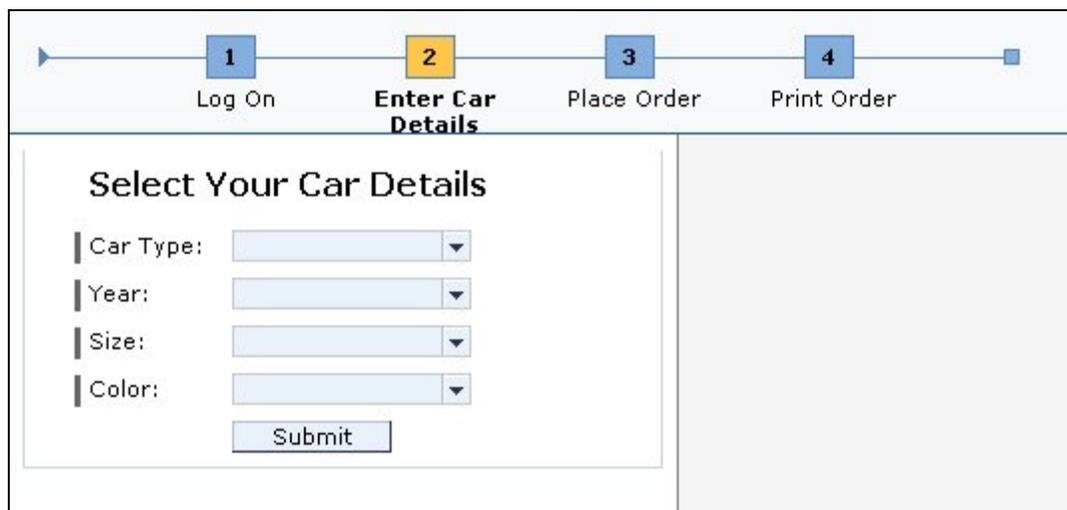
This article describes a model that implements the Signal Name as Event feature in a number of cases. It is a simple illustration that points out the benefit of this feature. I describe the specific use case of the model, explain the model logic, and then point out how Signal Name as Event is used.

The Use Case

A small-business owner has been leasing a number of cars for his business for 10 years. She has worked all this time with the same car agency, which has streamlined its operations to enable on-going customers to log onto the agency database, search for a new car to lease, and then find what is available, where, and how much the lease costs.

Steps in this Scenario

1. The customer logs on to the leasing agency database and her credentials are verified. Once this happens, the system displays the **Enter Car Details** view.

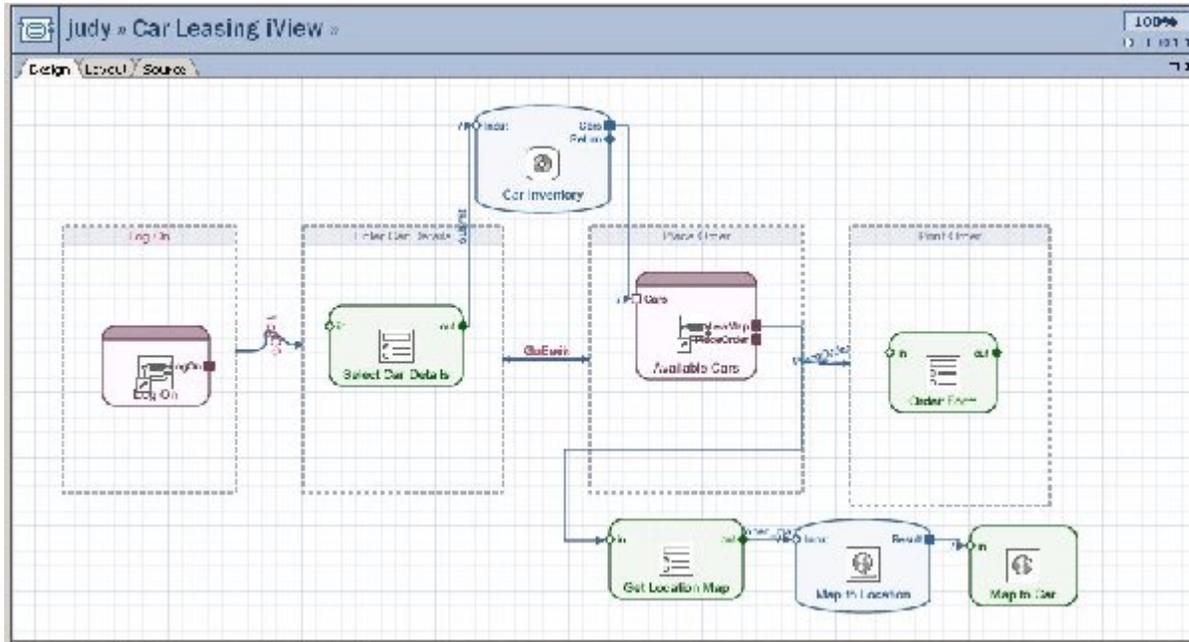


2. In the **Enter Car Details** view, the customer fills in a set of dropdown lists to select the general details (year, size, color, type of motor, mileage and so on) of the car she wants, and then clicks **Submit**. The *Submit* event sends the information to the Agency Inventory data service and also triggers the transition to the **Place Order** view.
3. The Agency Inventory data service sends the results (including car price and location) to a table in the **Place Order** view, showing all available cars that match the customer selections.

- The customer then reviews the details in the formal order form, and clicks *Print* to print out the order form, sign it and transfer it to the branch from which the car will be leased.

The Model – Overall Description

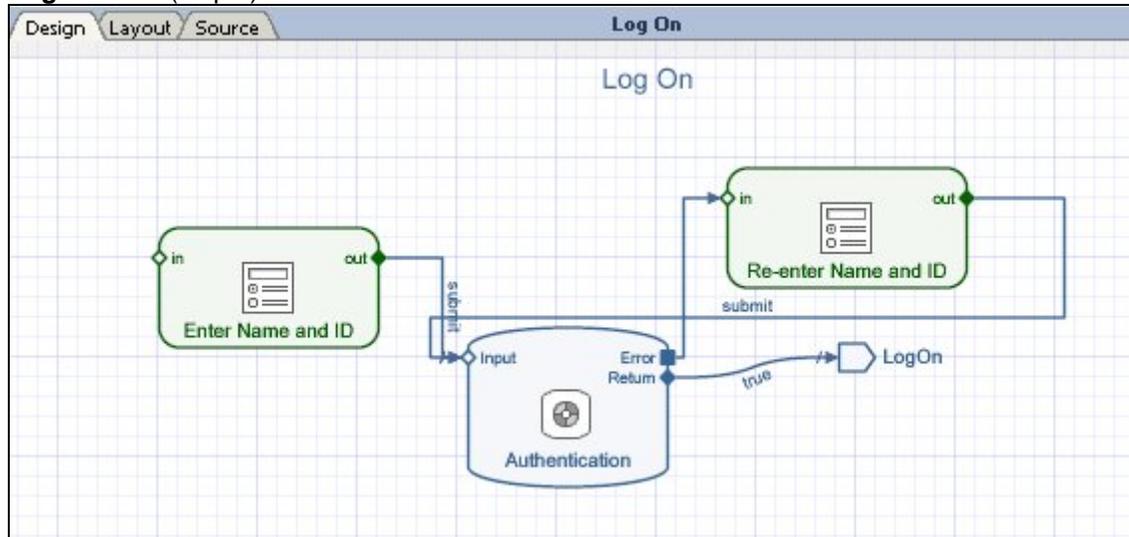
The model on the Design board is a four-step wizard that looks like this:



- Step 1 - **Log On** to the leasing agency system.
- Step 2 – **Enter Car Details**: A view that enables the user to select from a list of attributes to define the type of car wanted.
- Step 3 - **Place Order**: The user views the returned list of cars that match the attributes submitted and selects the car he/she wants to lease, including checking the location of the selected car.
- Step 4 – **Print Order**: The user prints out the order of the selected car, which serves as a preliminary order form for it at the agency.

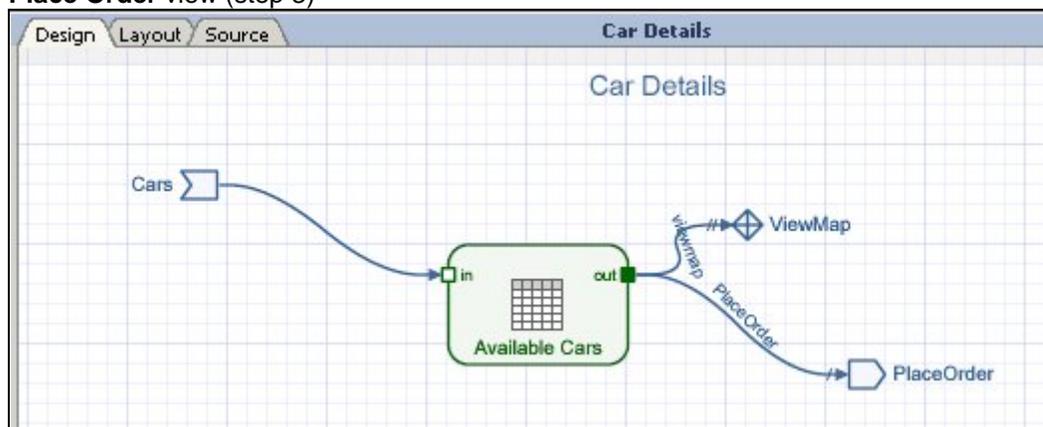
Implementations of Signal Name as Event:

- **Log On** view (step 1)



This is a reusable nested iView that enables customers to fill in a form with their User Name and ID, and have a data service of user names verify the log on. When the credentials are verified, a *True* signal is sent to the *LogOn* out signal / Out port of the view, and the *LogOn* transition is triggered, displaying the second – **Enter Car Details** – view.

- **Place Order** view (step 3)



This nested iView receives parameters returned from the Car Inventory web service, and channels the data from the nested iView port / Input signal to a table, which listed rows of data matching the customer requirements. This table has a double functionality:

- The customer can select a View Map icon on any line to display a popup window containing a URL iView that shows a map to the city listed in the *Location* column of the table. It contains a Close icon so that the customer can retain the table and check out a number of different locations.
- When the customer selects a line of the table that contains the data that matches her requirements, she clicks the *Place Order* button, which sends the *Place Order* signal to the *PlaceOrder* port, thereby triggering the transition to the final, **Print Order**, view.

In summary,

This model illustrates how the new Signal Name as Event feature – introduced in the Visual Composer SP Stack 14 patch – can connect between nested iViews, and thereby enable the reuse of certain of these model components. By naming an out port of a nested iView with the same name as an event, you can send parameters between different nested iViews within a containing iView, “reusing” iViews as standardized nested elements.

Copyright

© 2008 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, System i, System i5, System p, System p5, System x, System z, System z9, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, POWER5+, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

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.