

Service Prototyping Using the Service Component Wizard

Applies to:

Visual Composer for SAP NetWeaver Composition Environment 7.1

Summary

The service component wizard allows the modeler to model Visual Composer applications UI first. This means that the application can be modeled before the data services are ready.

Author(s): Jarrod Williams

Company: SAP

Created on: 25 December 2007

Author Bio



Jarrod Williams is a Visual Composer Solution Expert working in the Visual Composer Solution Office.

Table of Contents

Overview	3
Creating the Service.....	4
Consuming the Service	8
Configuring the Service.....	10
Summary	11
Related Content	11
Copyright.....	12

Overview

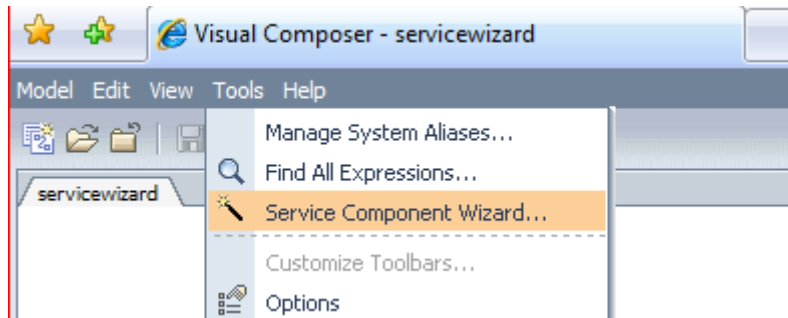
The service component wizard allows the modeler to model Visual Composer applications UI first. This means that the application can be modeled before the data services are ready. Refer to the article by Elizabeth Gutt [Visual Composer Modeling - From Prototype to Full-fledged Application](#) for an overall explanation of the prototyping concept in Visual Composer and especially to section 2 of the article.

With this new wizard, you can now create a full Visual Composer application while focusing on the UI and the flow of the application and basing it on demo (simulated) services. The demo services include dummy data and have input and output ports with corresponding fields to simulate the real services. This way, the developer can receive the application from the modeler, understand what the services do, and replace the demo services with the real back-end data services when they are ready.

You can create the demo services manually or by copying dummy data from a spreadsheet using the new service component wizard, as described in the following example:

Creating the Service

1. Open the wizard by choosing Tools -> Service Component Wizard.



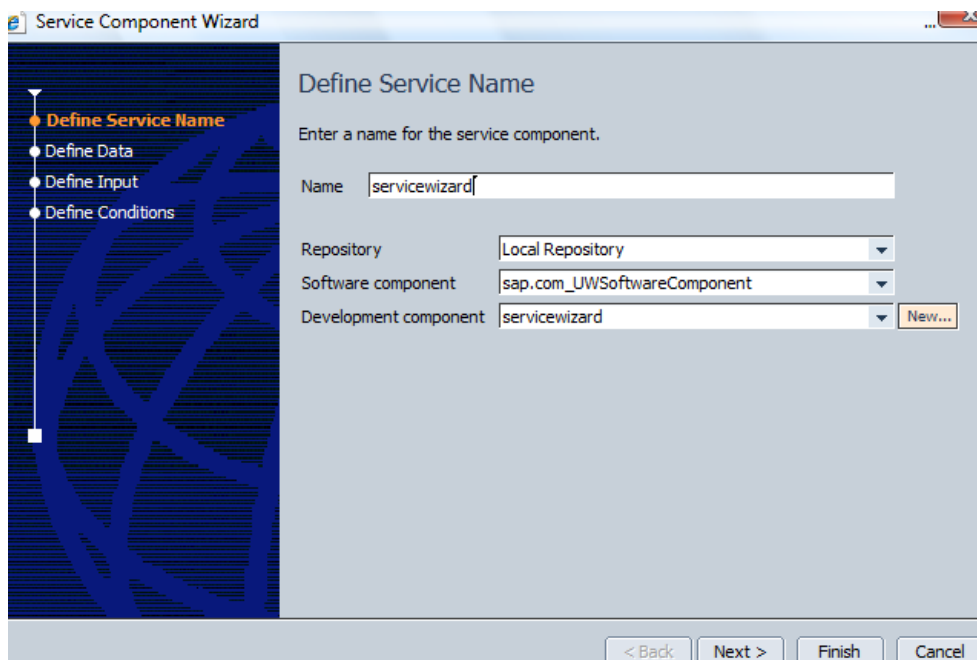
- a. The wizard can also be opened by dragging the Service Component icon from the Compose task panel and choosing the wizard from the displayed menu (in this case, the service is created automatically in the same development component as the model).

Operators

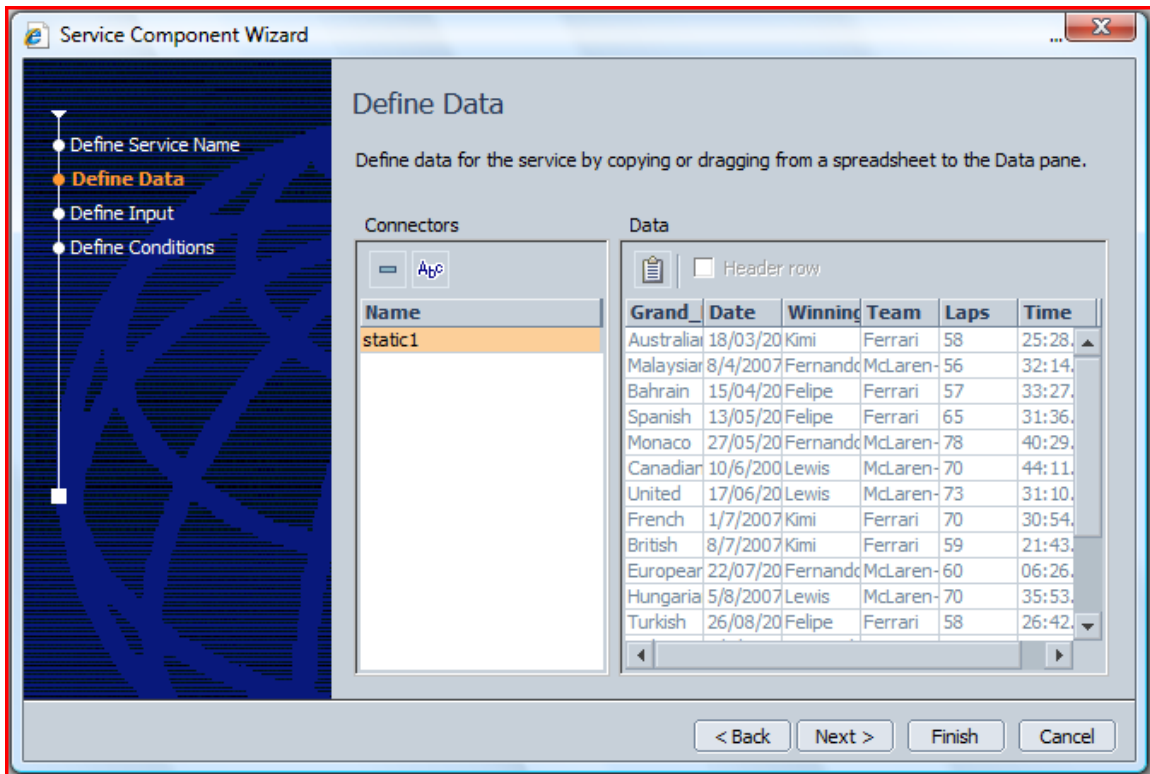


The rest of the steps are the same.

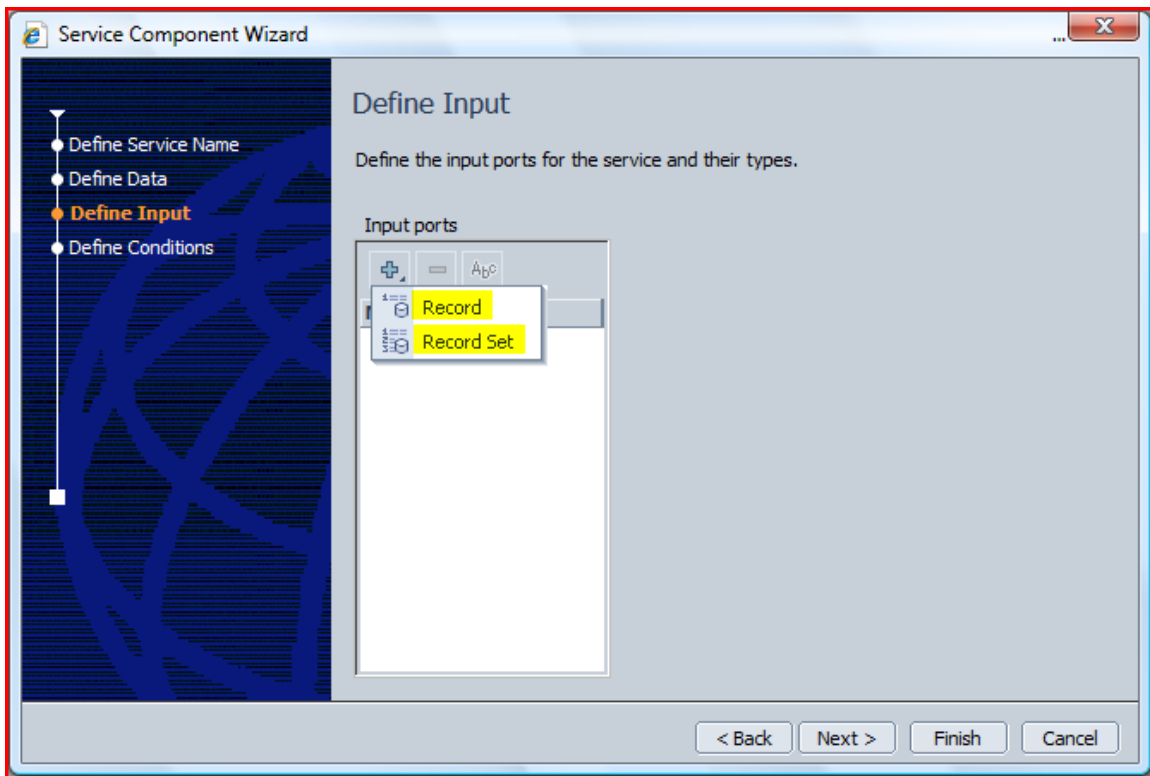
2. Enter the required information, saving the service component in the same development component as the model (to be able to use it in the local repository).



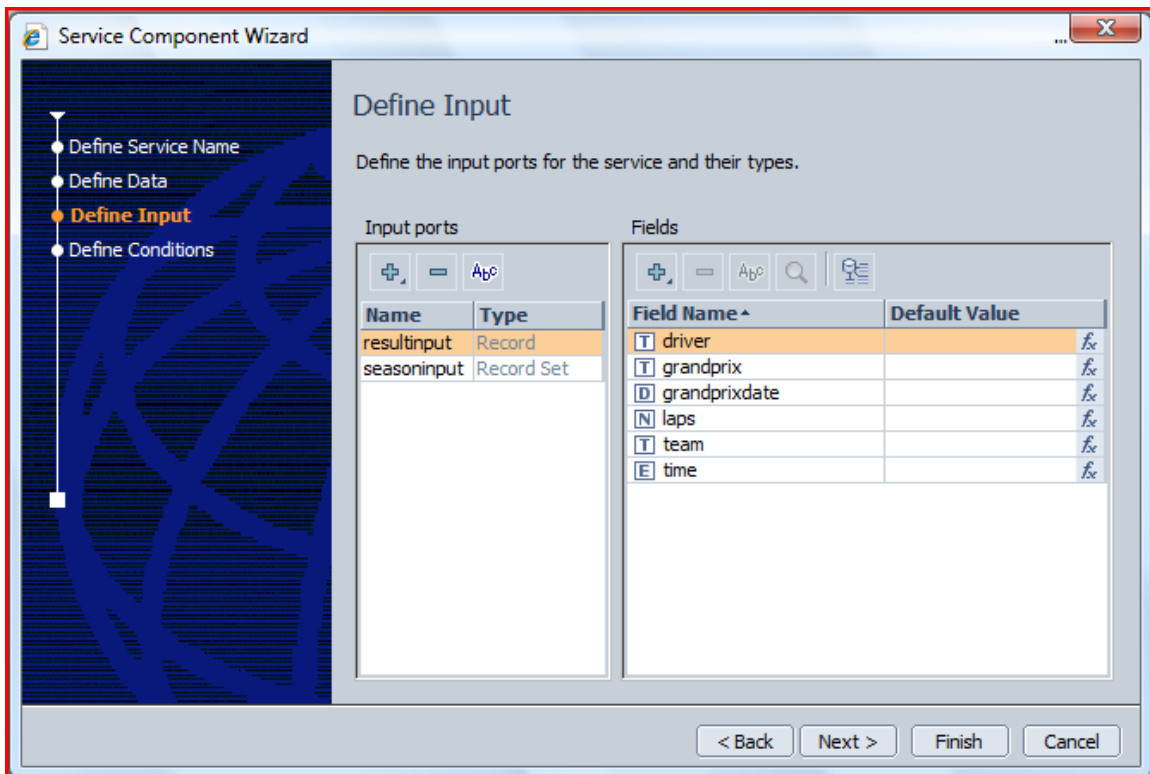
- Copy and paste or drag and drop the data from a spreadsheet; you can create the data and structure according to the needs of your application or service. In the screen below, you can see the data that was pasted from a spreadsheet containing the results of the 2007 Formula 1 season. You can see that the table includes data of different types: text, date and numeric. The original Excel file can be found attached to this document.



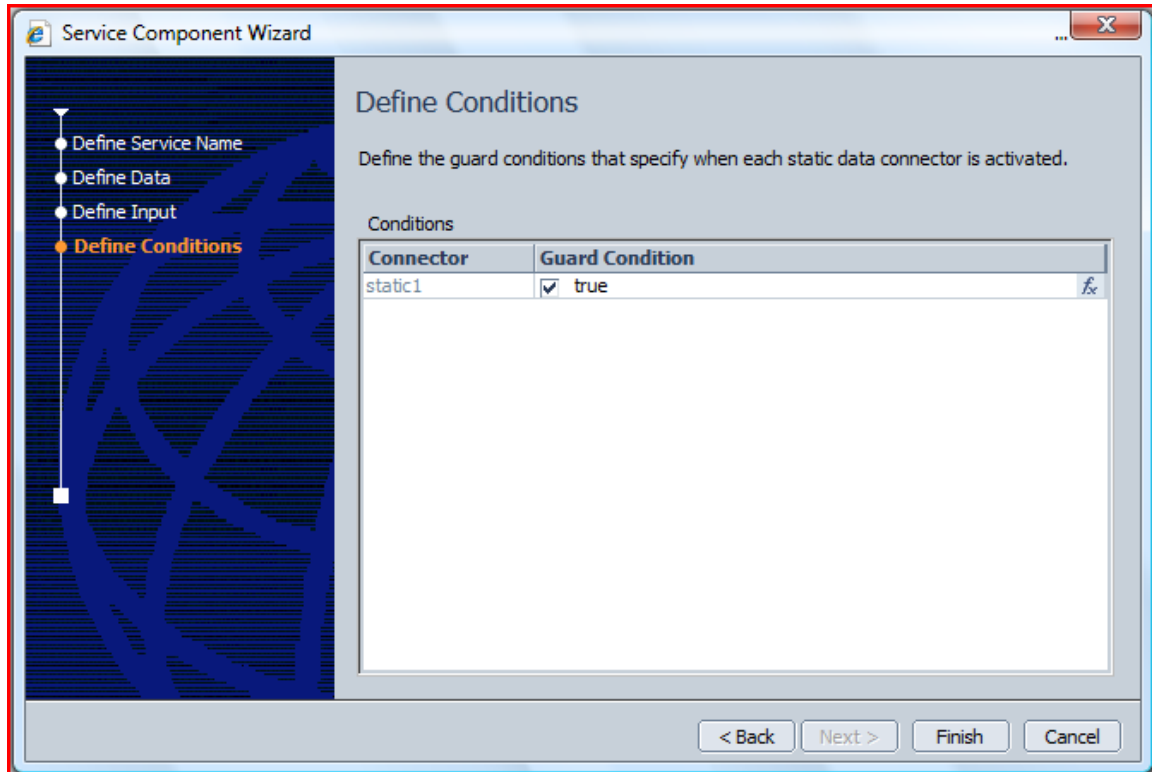
- The data is stored in a static data connector (static1); you can add additional static data connectors to your service by repeating the copy and paste operation from the spreadsheet. Having more than one static data connector in your service enables you to manipulate the data and display different results according to conditions you define in the last step of the wizard.
- In this step, we set up the input for the service. We can define either a form or a table as input. Record means that the input will be a form and Record Set means that the input will be a table.



6. In this example, inputs have been defined for a form and a table; one where the fields are entered per race and the other where fields are entered per season. Notice that the left pane indicates the input ports, whereas the right pane indicates the input fields for each of the input ports. We added the input fields manually using the Add (+) button.



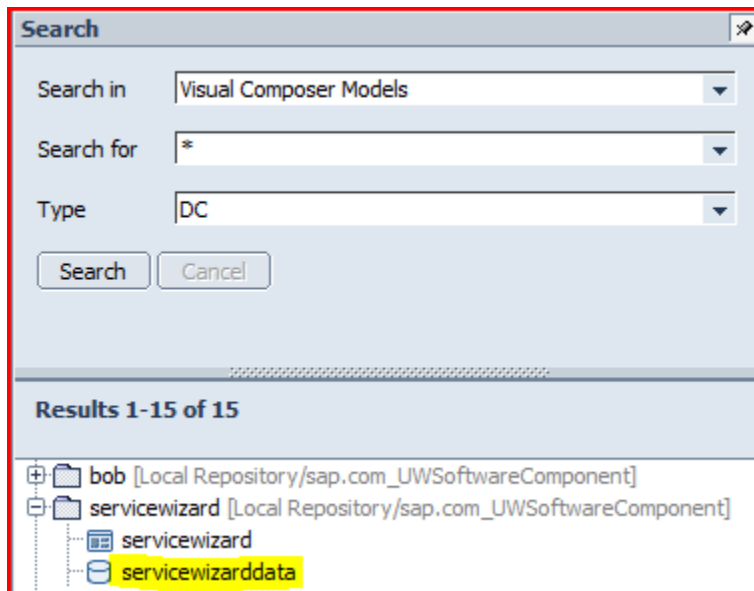
7. In the Define Conditions step, you can specify guard conditions that define when the different static data connectors are activated.



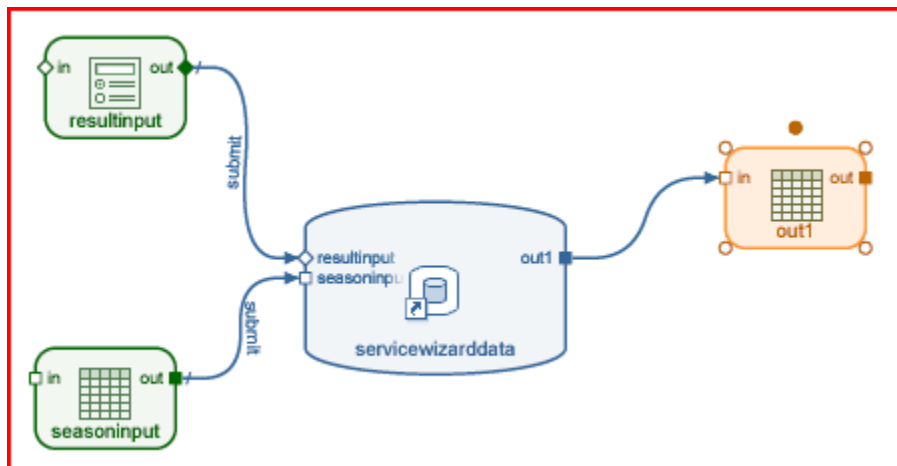
8. After the wizard is finished, you can see the modeled representation of the service on the Design board. The service component can then be saved and reused in the modeling of the application UI, and further manual changes can be made to the service as required by editing the model.

Consuming the Service

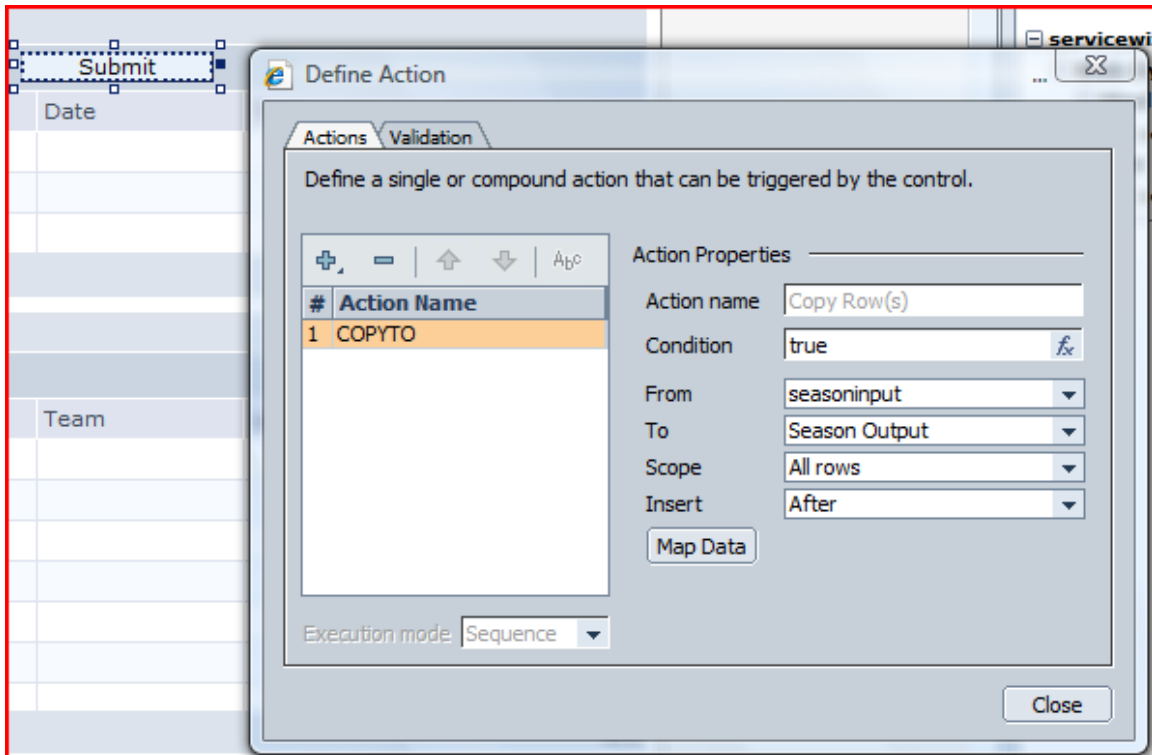
1. Create a composite view in Visual Composer.
2. Search for the service in the Search task panel, under Visual Composer models. The screen below shows that we have found the service component and it is ready to be dragged onto the model in which you want to consume it.



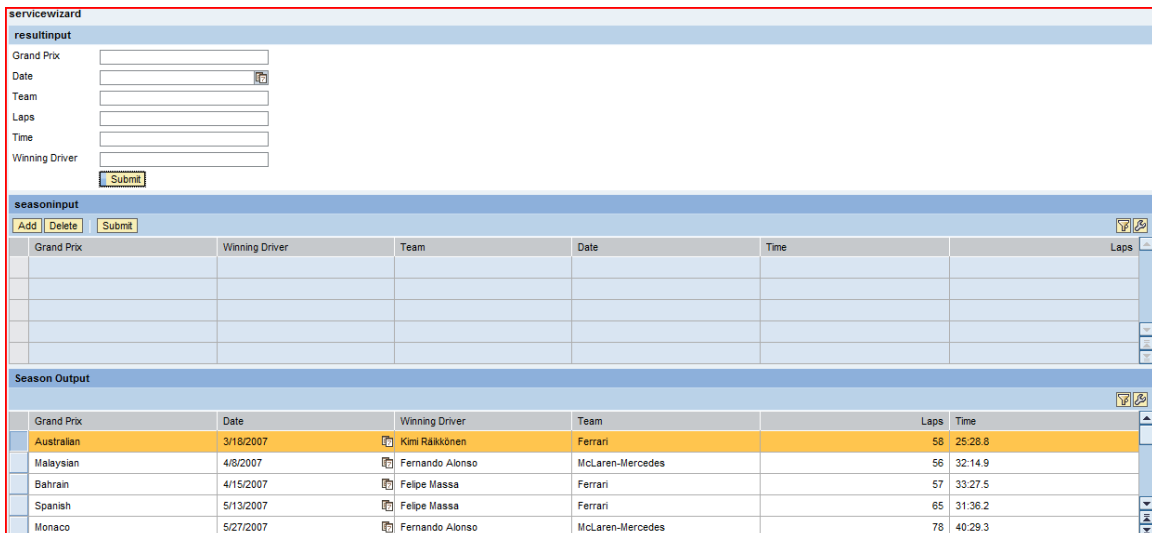
3. In the screen below, you can see that the service has the input and outputs ports we defined using the service component wizard. The service and the initial data can be used and manipulated like any other data service, the only difference is that changes made to the data are not saved and the data reverts to its original state when a new session is started. If you need to edit the service to add new ports or add more data, you can double-click the service to open it for editing on the Design board.



- Note that clicking the Submit button on an input form or table does not add the data to the service itself, because this data is initial data. To simulate adding data to the service, the data needs to be inserted into the output table using the Copy Row(s) action. In this example, the Copy Row(s) action is defined for the table input, specifying that all the data rows from the input table should be copied to the output table.



- At runtime, you can see that the input form and table are displayed and that the dummy data is returned when initializing the service by clicking the Submit button.



Configuring the Service

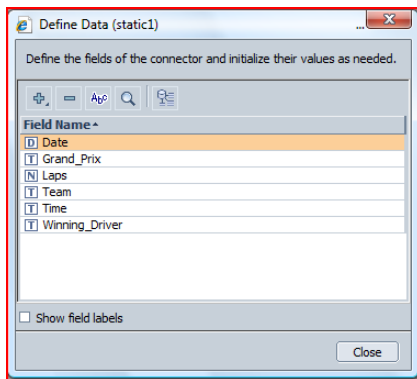
At any stage, you can edit the service manually in Visual Composer. You can redesign the input and output ports and add more dummy data to it.

Open the service by either double-clicking the service component icon on the Design board of the model or choosing the service component in the Open Model dialog box.

The service appears similar to this:



The input and output ports can be edited by right-clicking the port and then choosing Define Data; you can add and remove fields in this dialog box.



The demo data can also be edited by right-clicking the static data connector, choosing Define Data from the context menu and then clicking the Initialize Data button.

Summary

This article provided a basic example that demonstrates how you can use the service component wizard to create a demo service that returns a list of static data.

In addition to this, you can enhance the basic service, either by using the wizard or manually by editing the service, and add input ports, output ports, and conditions to define when to display certain data – according to the runtime user's input values.

Related Content

- [Visual Composer Modeling – From Prototype to Full-fledged Application](#)
- [SAP NetWeaver Visual Composer FAQ](#)
- [Visual Composer for SAP NetWeaver 7.1 page on SDN](#)

Copyright

© Copyright 2007 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.