

Generate an IMS feed using IMS Web 2.0 Solution for InfoSphere MashupHub

Skill Level: Intermediate

[Evgueni Liakhovitch \(evgueni@us.ibm.com\)](mailto:evgueni@us.ibm.com)
Software Developer, IBM®

[Yee-Rong Lai \(ylai@us.ibm.com\)](mailto:ylai@us.ibm.com)
Information Developer, IBM

[Shahin Mohammadi-Rashedi \(shahin@us.ibm.com\)](mailto:shahin@us.ibm.com)
IMS™ SOA Demonstration Team Lead

IBM Corporation
22 May 2009

Abstract

This tutorial takes you through the steps to generate a feed from an IMS transaction and to incorporate the feed into a Web 2.0 application by using IBM Rational® Developer for System z® and IBM Mashup Center.

About this tutorial

This tutorial shows you how to use the Enterprise Service Tools (EST) wizard in IBM Rational Developer for System z and IBM Mashup Center to transform and modernize your IMS applications.

IBM Mashup Center includes two components:

- InfoSphere™ MashupHub is a browser-based visual tool for creating, storing, transforming, and remixing feeds to be utilized in mashup and situational applications.
- Lotus® Mashups is a browser-based visual tool for mashing information from various sources into a single view of information to create a flexible and dynamic application. IMS feeds and other widgets can be arranged to create a rich user interface.

In this tutorial, you will learn how to:

- Use the EST wizard to generate the service artifacts, including the COBOL converter and the XML correlator from the IMS application source file
- Enable the IMS application as a service feed provider running on InfoSphere MashupHub
- Use Lotus Mashups to create a Web application from IMS feed and other widgets

Objectives

To understand and gain hands on experience creating web applications from IMS assets using IBM Web 2.0 solutions. Web 2.0 enables users with little to no programming experience to create applications using the Web itself as a platform, therefore reducing development time and education costs.

Upon completion of this study, you will be able to:

- Use Rational Developer for System z and its EST wizard to generate the required service artifacts for an IMS Web 2.0 project
- Use InfoSphere MashupHub and Lotus Mashups to create and transform an IMS feed and to create a dynamic Web application.

System requirements for the tutorial:

- Software installed on Windows®
 - Rational Developer for System z Version 7.5
- System software installed on IBM z/OS®
 - IMS Version 10
 - IMS Connect Version 10 with XML Adapter configured
 - OTMA
 - TCP/IP

Checklist for first-time implementation

You may find it helpful to have the following checklist available before proceeding with your own implementation for the first time.

A tutorial checklist is provided for this exercise.

	Your environment	This tutorial:
COBOL copybook	This can be obtained from IMS application programmers.	C:\IMS Web 2.0 PhoneBook \IMSPHBW.cpy
IMS Connect host name (or IP address) and port number	This can be obtained from IMS system programmers.	Host name: ZSERVEROS.DFW.IBM.COM Port number: 9999
IMS data store	This can be obtained from IMS system programmers.	IMSC
Workspace directory and project name will be used by Rational Developer for System z when generating artifacts.	A naming standard is recommended.	Workspace directory: C:\Workspaces7.5\SANDBOX Project name: SampleFeedApp

Table 1. Checklist for information and files you need for implementation

Overview of development tasks

To complete this tutorial you will perform the following tasks:

- **Task 1. Create an IMS Feed**
 - **Step 1.** Generate the XML converter driver and the correlator file by using the EST wizard in Rational Developer for System z
 - The XML converter driver is generated based on the IMS application source file that describes the input and output language structure. The driver must be deployed to IMS Connect. IMS Connect has an XML adapter function that converts the request and response messages between XML and bytes based on the specified XML converter driver.
 - The correlator file contains information about the transaction (such as transaction code and XML converter driver name) as well as a list of input parameters that the IMS application accepts. InfoSphere MashupHub needs this information in order to communicate with IMS and the application.
 - **Step 2.** Deploy the XML converter driver file
 - **Step 3.** Enable the IMS feed service using the InfoSphere MashupHub

Generate an IMS feed using IMS Web 2.0 Solution for InfoSphere MashupHub

© Copyright IBM Corporation 2009. All rights reserved.

- **Task 2. Create a mashup.** Use Lotus Mashups to create a Web mashup from the IMS feed and other widgets
 - **Step 1.** Adding an IMS feed to Lotus Mashups
 - **Step 2.** Adding components (IMS feed and other widgets) to a mashup page
 - **Step 3.** Wiring the components on the page
- **Optional Task** – Modify IMS feed to improve user experience

Figure 1 shows how various tools interact to help you accomplish the tutorial tasks.

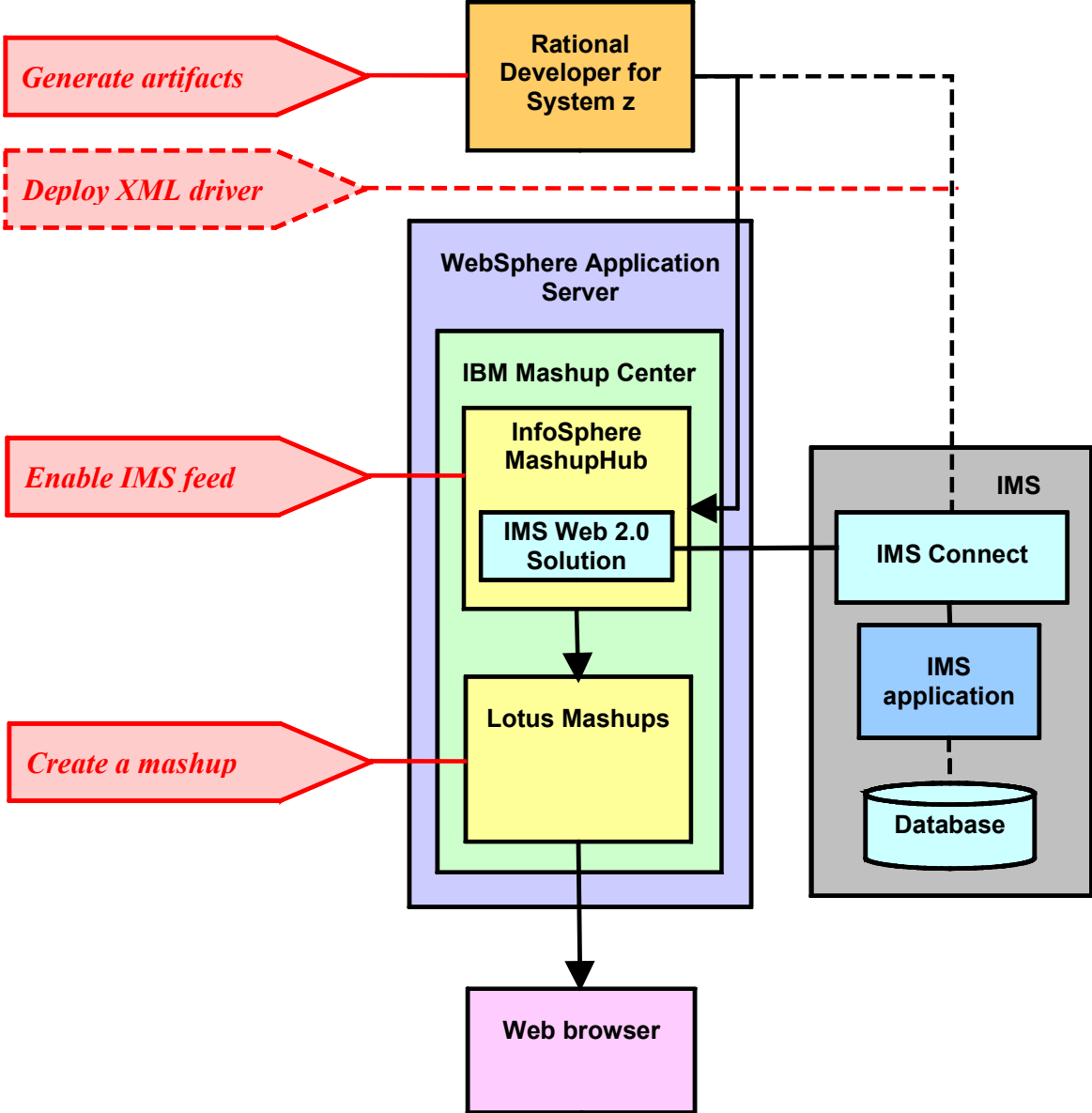


Figure 1. Using Rational Developer for System z and IBM Mashup Center to accomplish lab objectives

Task 1. Create an IMS Feed

Step 1: Generating the XML converter driver and the correlator file by using the EST wizard in Rational Developer for System z

1- 1. Using Rational Developer for System z and the Enterprise Service Tools perspective

1.1 The IBM Rational Developer for System z V 7.5 is started and you are using the Workspaces7.5 \SANDBOX

Important!

You are using `C:\Workspaces7.5\SANDBOX` as your work workspace.

What is a workspace?

A workspace is a directory that stores files for your projects. You can select your own directory or take the default directory. Artifacts created by RDz will be stored in this directory.

1-2. Switching to the Enterprise Service Tools perspective

Switch from the default z/OS Projects perspective to the **Enterprise Service Tools** perspective. Within Eclipse, there are several ways to change perspectives.

1. From the **Window** menu, select **Open Perspective → Other**.

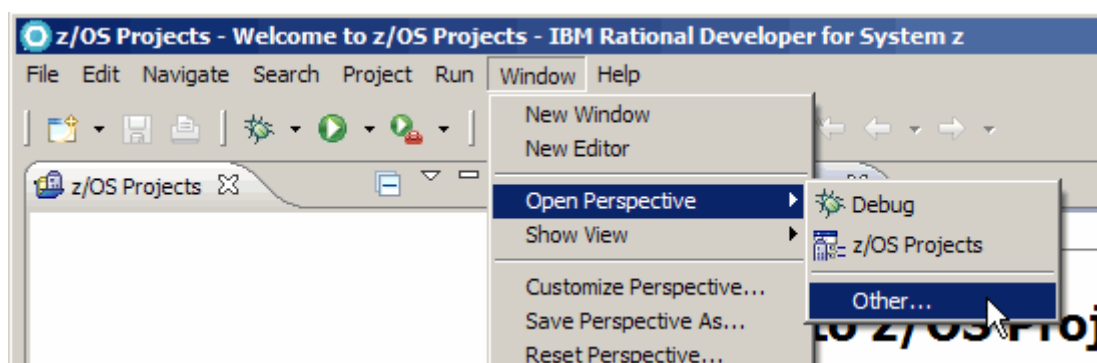


Figure 2. Opening a perspective in RDz

2. Select **Enterprise Service Tools** from the *Open Perspective* dialog box.

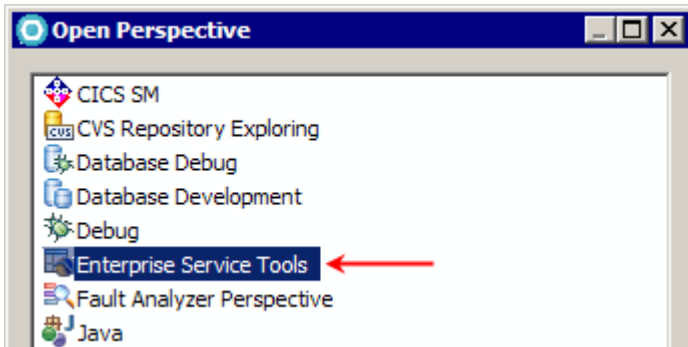


Figure 3. Choosing the Enterprise Service Tools perspective

3. Press **OK** to switch to the **Enterprise Service Tools** Perspective.

What is a perspective?

A perspective defines the initial set and layout of views in the Workbench window. Within the window, each perspective shares the same set of editors. Each perspective provides a set of functionality aimed at accomplishing a specific type of task or works with specific types of resources. For example, the Java™ perspective combines views that you would commonly use while editing Java source files, while the Debug perspective contains the views that you would use while debugging Java programs.

1-3. Creating a new IMS Web 2.0 project using RDz

1. From the EST **Project Explorer** window, right click and select **New** → **IMS Web 2.0 Project**.

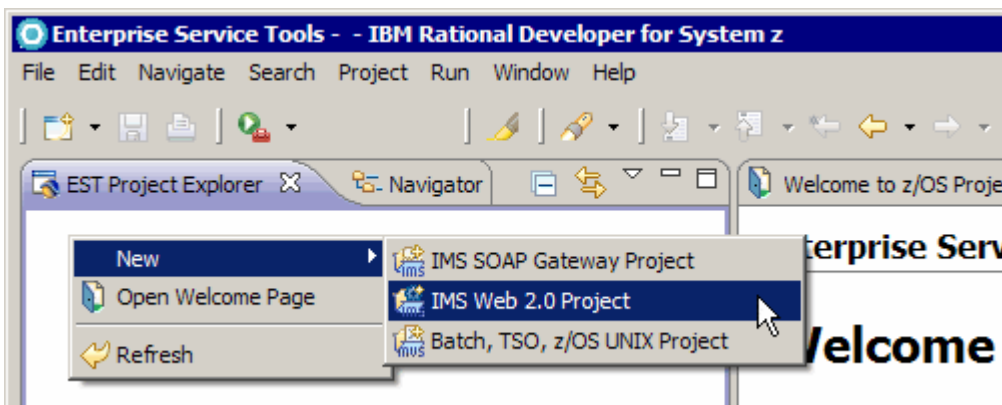


Figure 4. Starting the IMS Web 2.0 Project

What is Web 2.0?

Web 2.0 is the business revolution in the computer industry caused by the move to the Internet as a platform. Web 2.0 empowers users to manipulate data and combine various services into a single Web experience.

2. Specify the **Project name**:

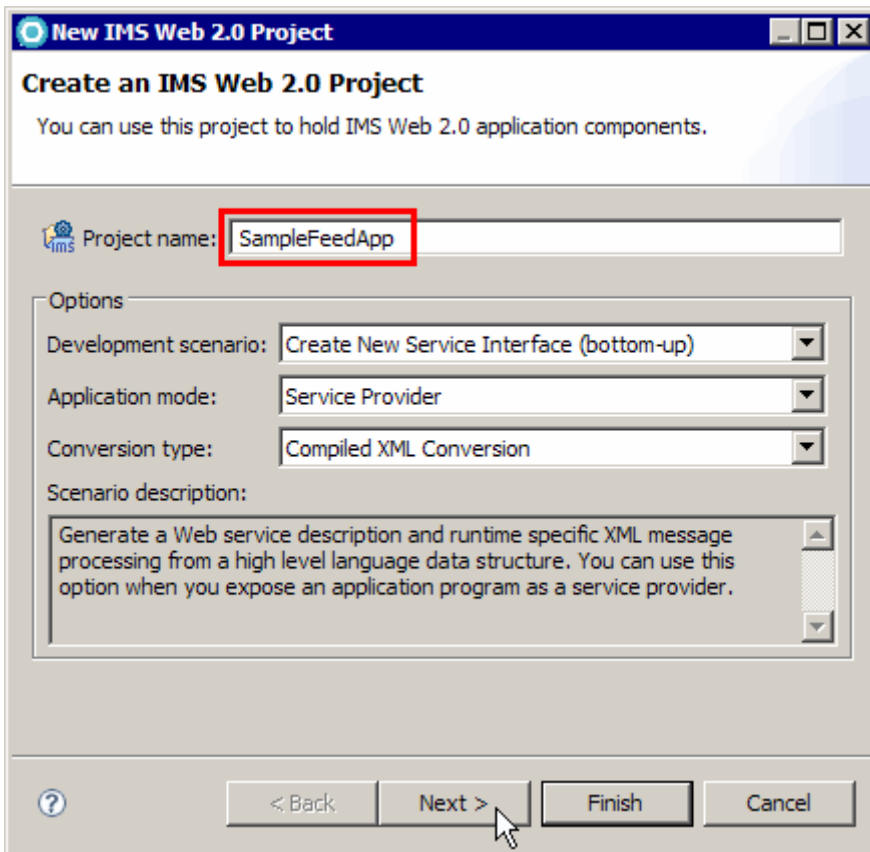


Figure 5. Specifying the project name

3. Accept the default options as shown in the above screen capture and click **Next**.
4. Import the source program for the generation. In this tutorial, import the source from the local file system by clicking **File System**.

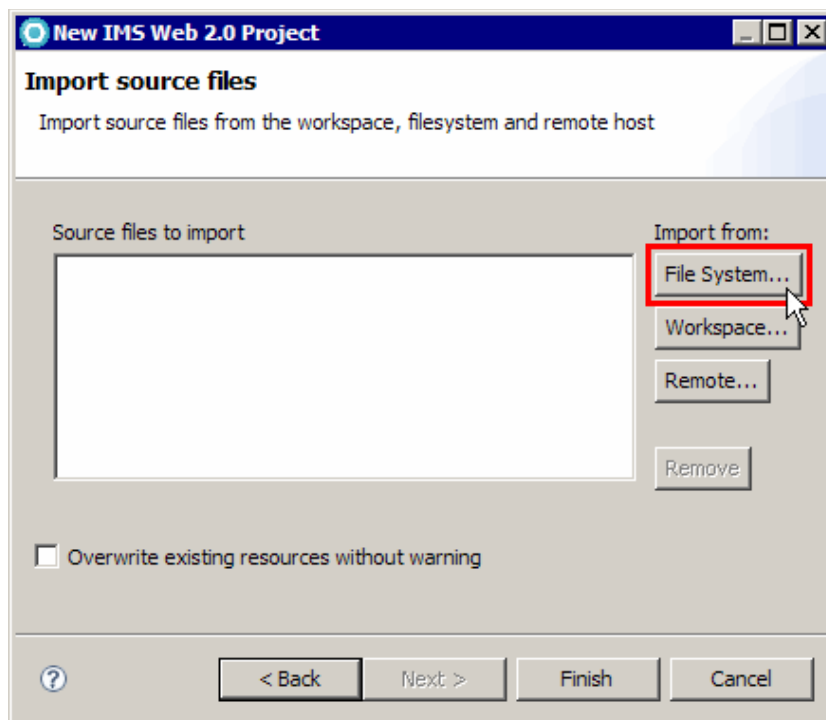


Figure 6. Importing source files menu

5. Find the file **C:\IMS Web 2.0 Phonebook\IMSPHBW.cpy** and click **Open**.

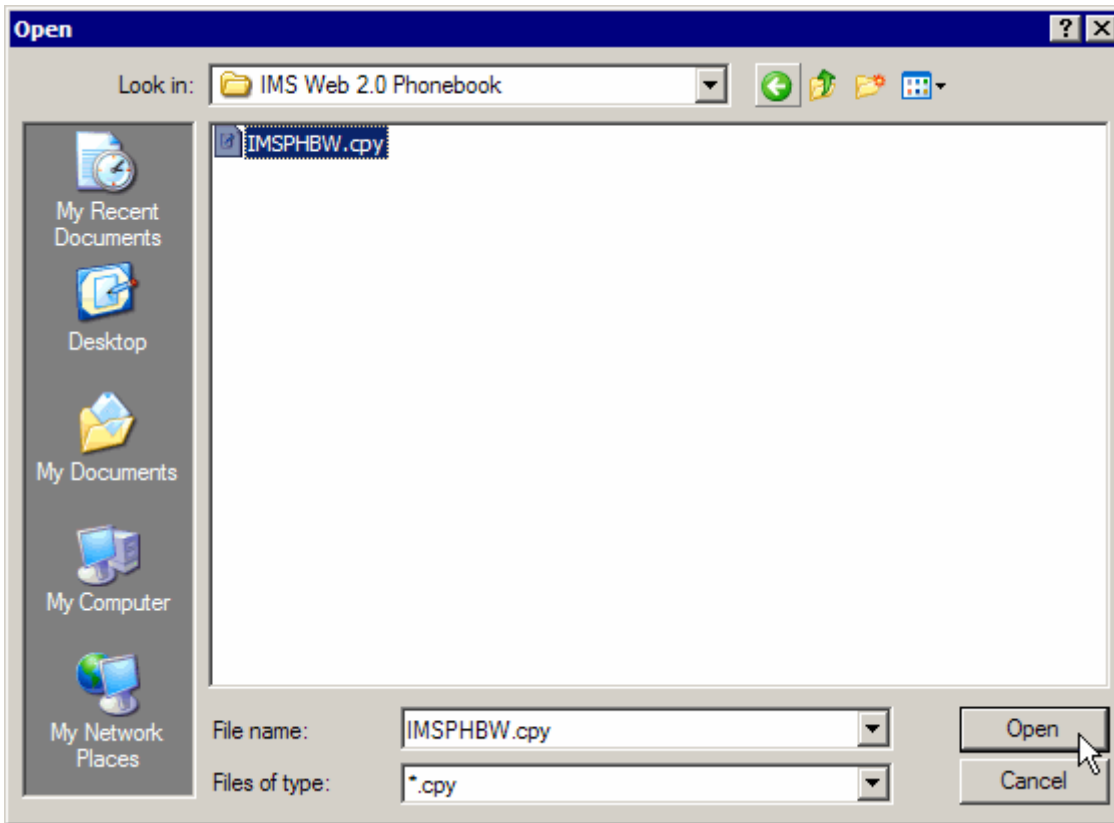


Figure 7. The Open file dialog

6. Click **Finish**, as shown in the following figure.

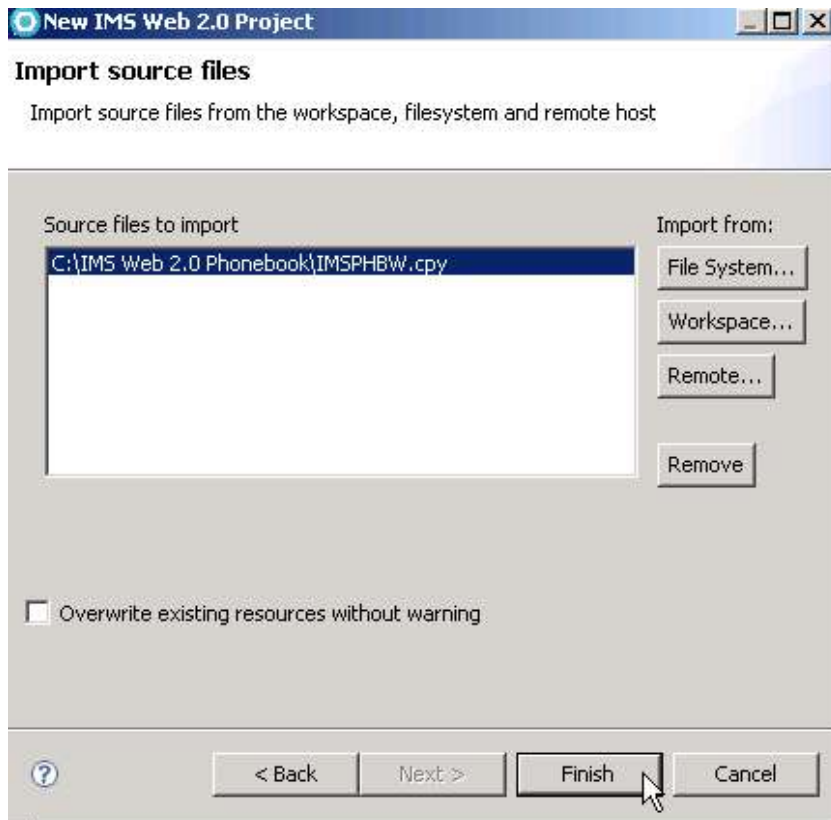


Figure 8. The Import source files menu

7. (Optional) In the **EST Project Explorer**, right click **IMSPHBW.cpy** and select **Open**, as shown in the following figure. This will open the COBOL data structure from which the service will be generated (as shown in Figure 10).

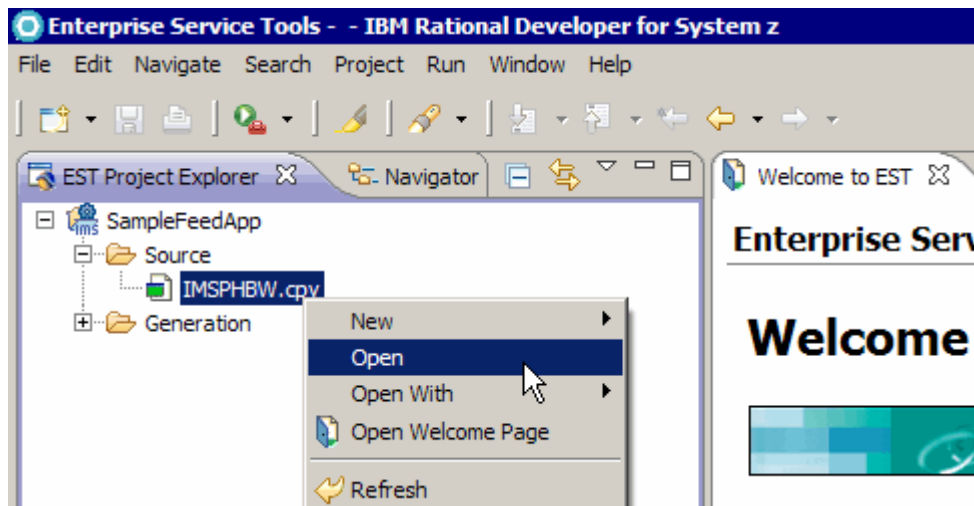


Figure 9. Opening the COBOL source file

```
Line 31      Column 1      Insert
+---+*A-1-B---+---2---+---3---+---4---+---5---+---6---+
01  INPUT-MSG.
    02  IN-LL          PICTURE S9(3) COMP.
    02  IN-ZZ          PICTURE S9(3) COMP.
    02  IN-TRCD        PICTURE X(10) .
    02  IN-CMD          PICTURE X(8) .
    02  IN-NAME1        PICTURE X(10) .
    02  IN-NAME2        PICTURE X(10) .
    02  IN-EXTN        PICTURE X(10) .
    02  IN-ZIP          PICTURE X(7) .

01  OUTPUT-MSG.
    02  OUT-LL          PICTURE S9(3) COMP VALUE +0.
    02  OUT-ZZ          PICTURE S9(3) COMP VALUE +0.
    02  OUT-MSG         PICTURE X(40) VALUE SPACES.
    02  OUT-CMD         PICTURE X(8) VALUE SPACES.
    02  OUT-NAME1       PICTURE X(10) VALUE SPACES.
    02  OUT-NAME2       PICTURE X(10) VALUE SPACES.
    02  OUT-EXTN        PICTURE X(10) VALUE SPACES.
    02  OUT-ZIP         PICTURE X(7) VALUE SPACES.
    02  OUT-SEGNO       PICTURE X(4) VALUE SPACES.
```

Figure 10. COBOL source

8. Right-click the project and select **Generate IMS Web 2.0 resources**.

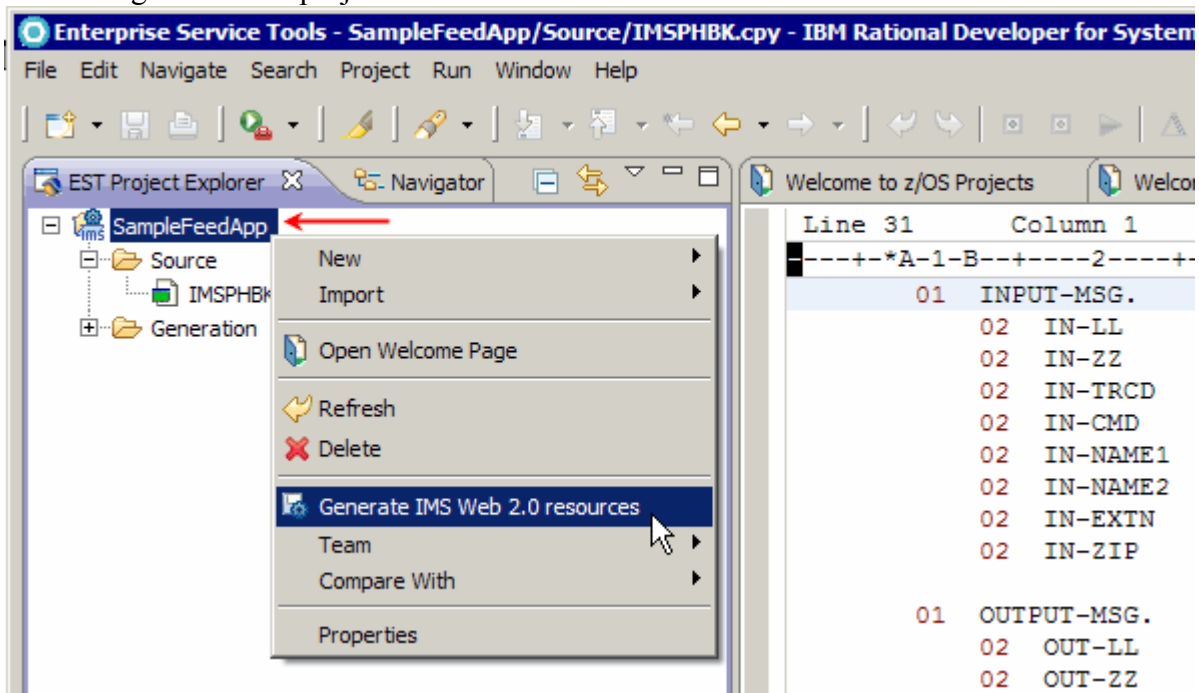
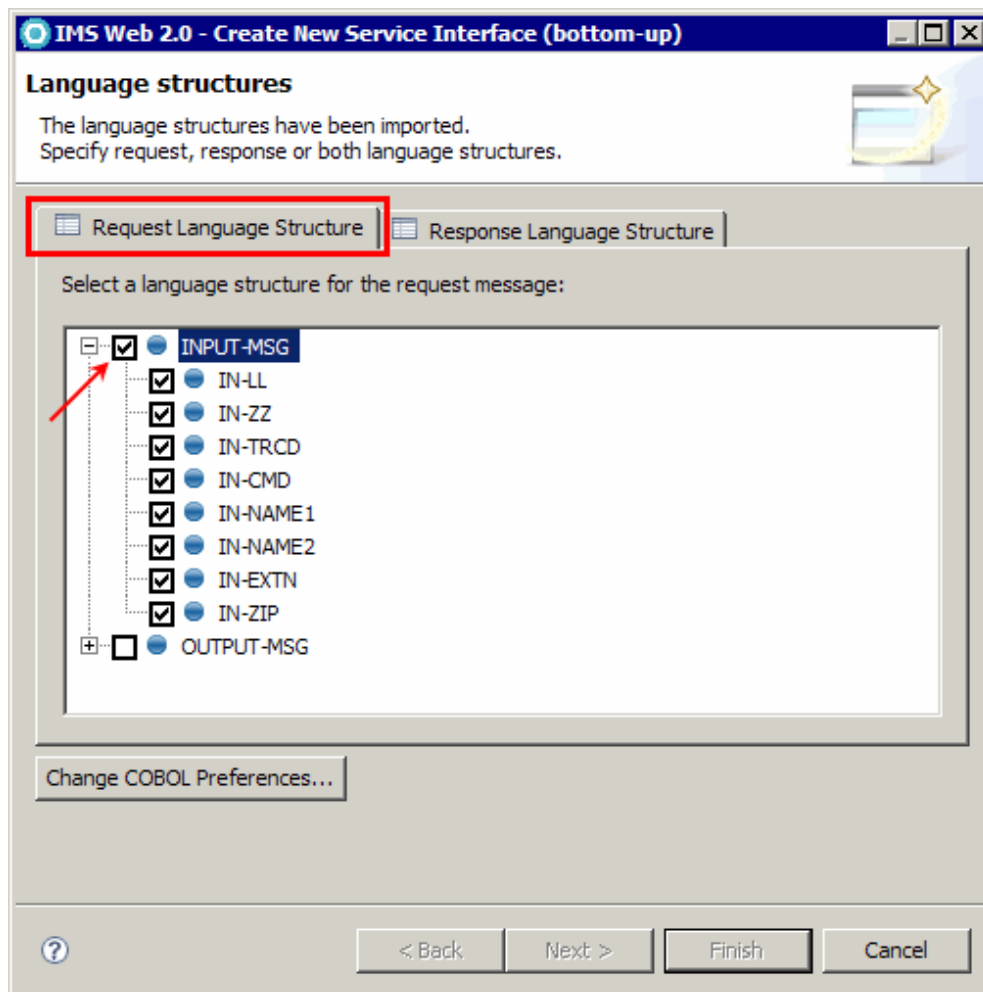


Figure 11. Selecting Generate IMS Web 2.0 resources

9. On the first page of the wizard, specify the data structures to be used for service request and response, as shown on Figures 12 and 13. Click **Next**.



Generate an IMS feed using IMS Web 2.0 Solution for InfoSphere MashupHub

© Copyright IBM Corporation 2009. All rights reserved.

Figure 12. Specifying the Request Language Structure

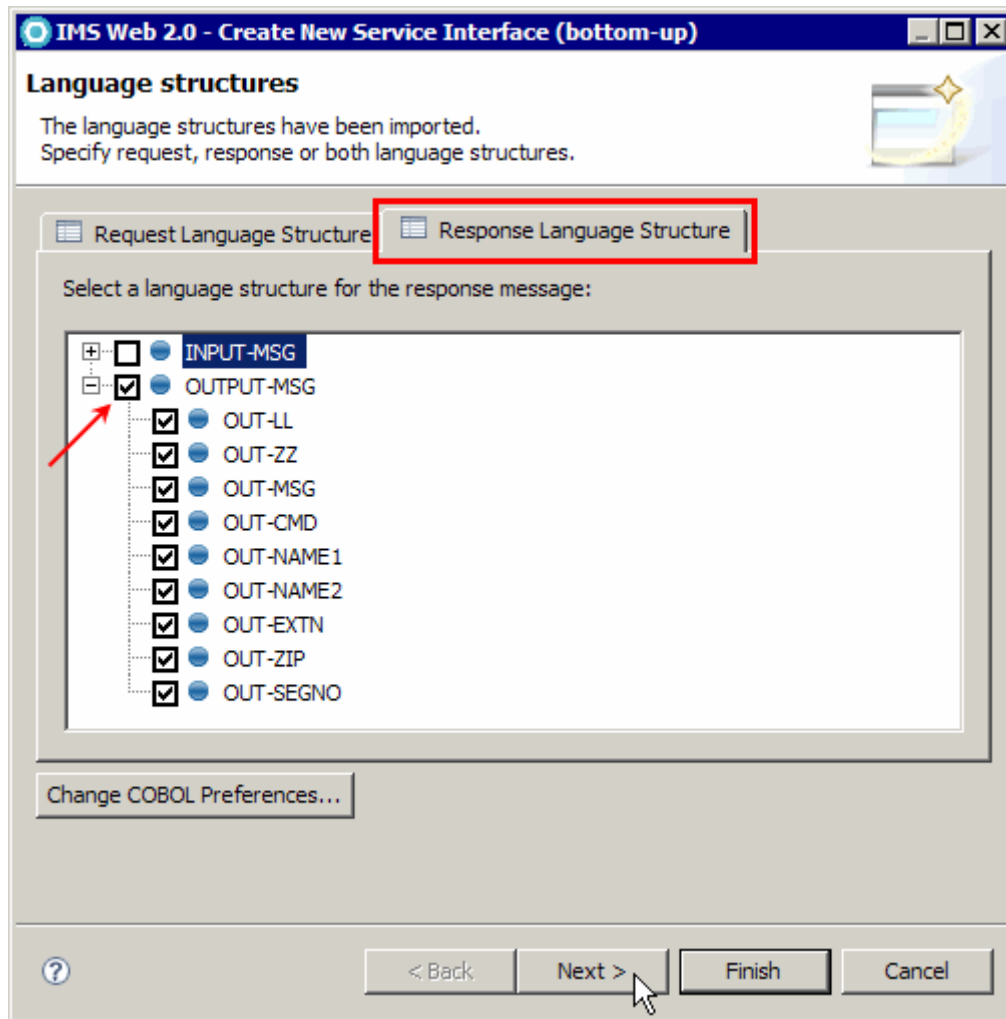


Figure 13 Specifying the Response Language Structure

10. In this tutorial, do not change any **Generation options** (Figure 14). Click **Next**.

IMS Web 2.0 - Create New Service Interface (bottom-up)

Generation options

Specify generation options for the Web Services enablement artifacts.

XML Converters | XML Schemas | **Advanced Options**

Specify identification attributes

Converter program name prefix: IMSPHBW

Author name: WD4Z

Service program name: IMSPHBW

Specify Enterprise COBOL for z/OS properties

Compiler level: 4.1

XMLPARSE option: COMPAT Enterprise COBOL for z/OS XML parser

Optimization

Specify character encodings

Request code page: 1208 Unicode, UTF-8

Host code page: 1140 USA, Canada, etc. EBCDIC with Euro

Response code page: 1208 Unicode, UTF-8

< Back | **Next >** | Finish | Cancel

Figure 14. The Generation Options screen

11. Specify the name of the transaction that provides the service (**IVTNO**). Click **Next**.

The screenshot shows the 'IMS Web 2.0 Service Provider' wizard. The title bar reads 'IMS Web 2.0 - Create New Service Interface (bottom-up)'. The main title is 'IMS Web 2.0 Service Provider'. Below the title, it says 'Specify properties for the IMS Web 2.0 Correlator file. This file is used by the IMS Web 2.0 runtime.' The wizard is divided into two main sections. The first section, 'Specify targets for the IMS Web 2.0 Correlator file', has 'Generate to:' with 'Same project' selected and 'Remote location' unselected. The 'File container:' field contains '/SampleFeedApp/Generation/Targets' and has a 'Browse...' button. The 'File name:' field contains 'IMSPHBW' and has a '.xml' suffix. There is a checked 'Overwrite' checkbox. The second section, 'Specify IMS Connect interaction properties', has 'Transaction code:' with 'IVTNO' entered and highlighted by a red box. The 'Socket timeout:' field contains '0' and is labeled '(in milliseconds)'. The 'Execution timeout:' field contains '0' and is labeled '(in milliseconds)'. The 'LTERM name:' field is empty. At the bottom, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'. A mouse cursor is pointing at the 'Next >' button.

Figure 15. Specifying the transaction code

Socket timeout is the time in milliseconds for the server to receive a response from IMS Connect before disconnecting the socket. A value of 0 indicates that you want an interaction to run without a time limit.

Execution timeout is the time in milliseconds for IMS Connect to send a message to IMS and receive the response. A value of 0 indicates that you want to the timeout value that is specified in the IMS Connect configuration member.

12. Do not change any file generation options for this tutorial. Click **Finish**.

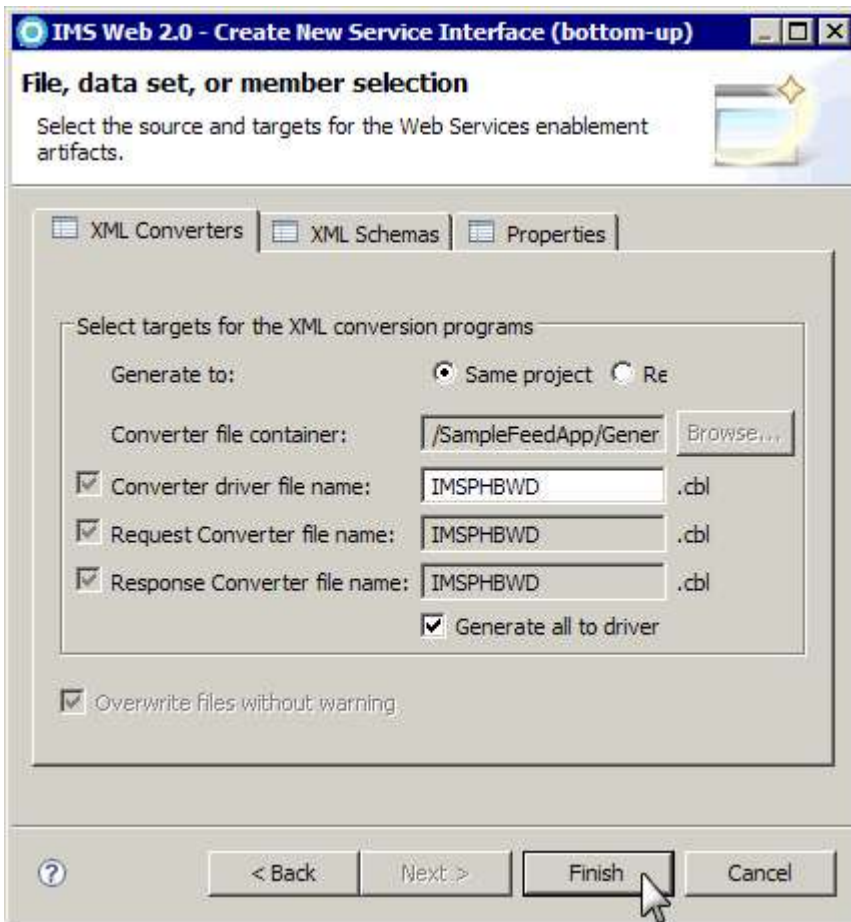


Figure 16. File generation options

The service artifacts are generated in the targets folder, as shown in the following figure.

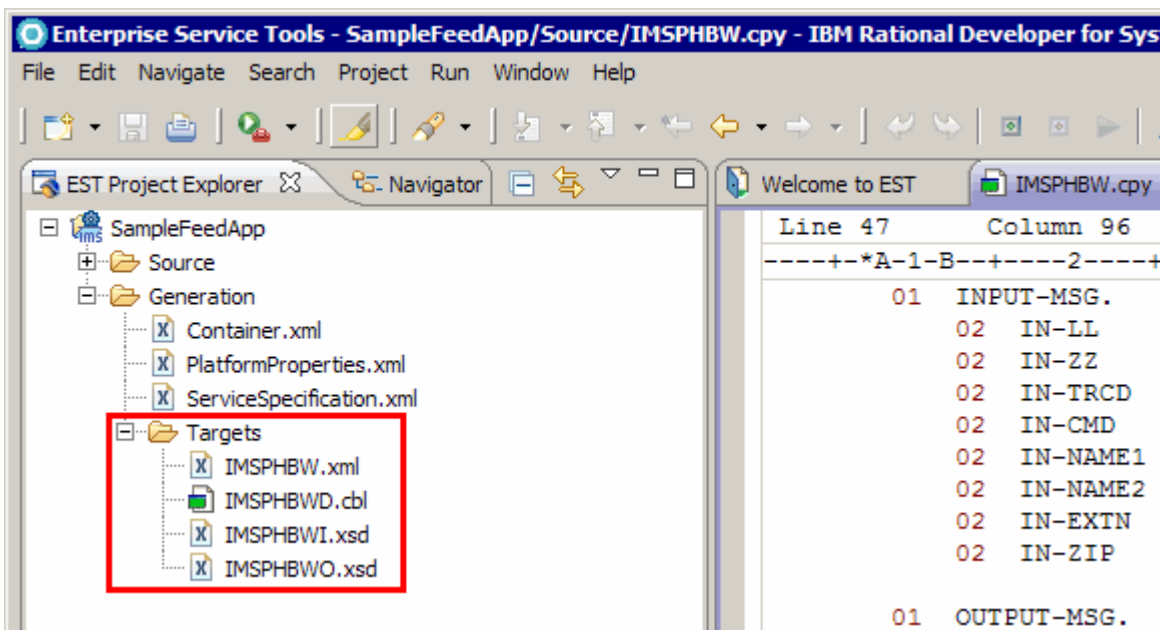


Figure 17. Service artifacts

IMSPHBW.xml is the XML correlator file. It is to be deployed to the **InfoSphere MashupHub**.

Generate an IMS feed using IMS Web 2.0 Solution for InfoSphere MashupHub

© Copyright IBM Corporation 2009. All rights reserved.

IMSPHBWD.cbl is the **XML converter** which is to be built and deployed to **IMS Connect**.

Step 2: Deploying the XML converter driver file

For this tutorial, the IMSPHBWD.cbl XML converter file has been deployed to IMS Connect for you, so no action is needed.

For your own implementation in your environment, you would need to:

1. Ensure that the XML adapter function in IMS Connect is properly configured.
2. FTP the XML converter driver to your host data set.
3. Compile and link the XML converter driver into a data set that is concatenated to the STEPLIB in the IMS Connect startup JCL.

Step 3: Enabling the IMS feed service using the InfoSphere MashupHub

1. Start **InfoSphere MashupHub**.
 - 1.1 Select **Start → Programs → IBM Mashup Center → Start Server**.
A console opens that displays informational messages about the status.
 - 1.2 After waiting for initialization status, the console indicates that the server is open for e-business, and you are prompted to press any key to continue. This may take a few minutes. Press **Enter**.
 - 1.3 Select **Start → Programs → IBM Mashup Center → InfoSphere MashupHub**. A Web browser will launch.
 - 1.4 If a security alert window pops up with some information about the security certificates and asks you if you want to proceed, click **Yes**.
A login screen for the **InfoSphere MashupHub** displays, as shown in the following figure.

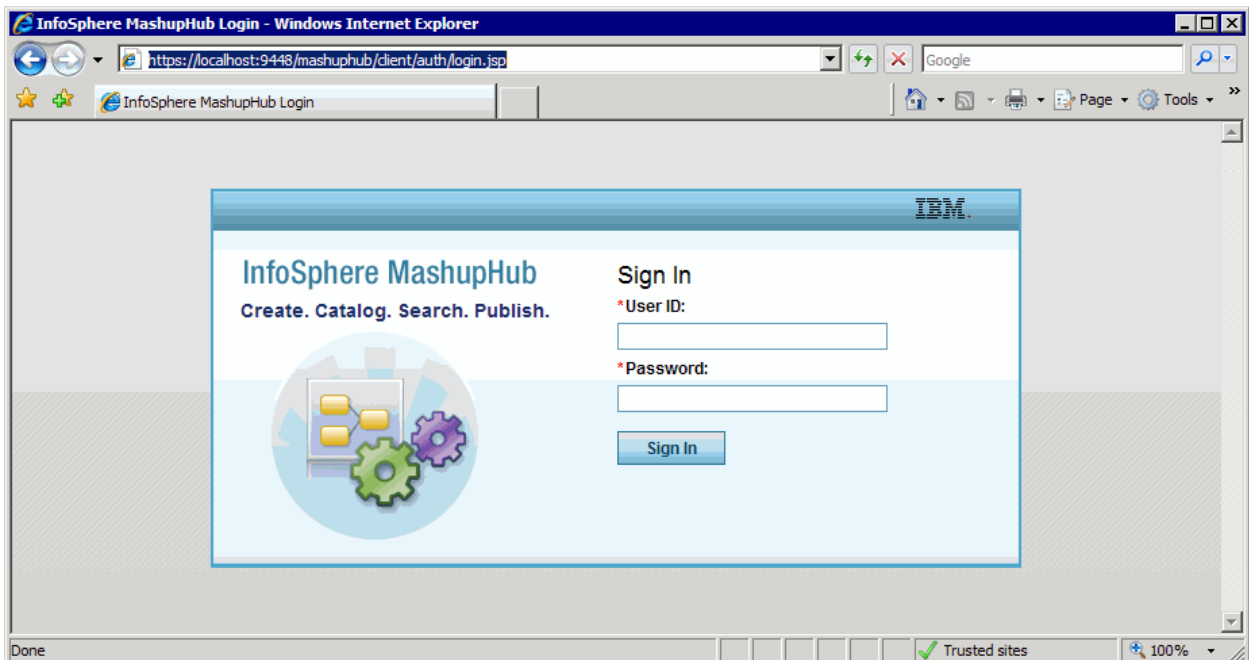


Figure 18. InfoSphere MashupHub login screen

2. Sign in using the following information:
 - User ID: `admin`
 - Password: `password`

Generate an IMS feed using IMS Web 2.0 Solution for InfoSphere MashupHub

© Copyright IBM Corporation 2009. All rights reserved.

3. Once signed in, select **New Feed** under the **Create** group, as shown in the following figure.

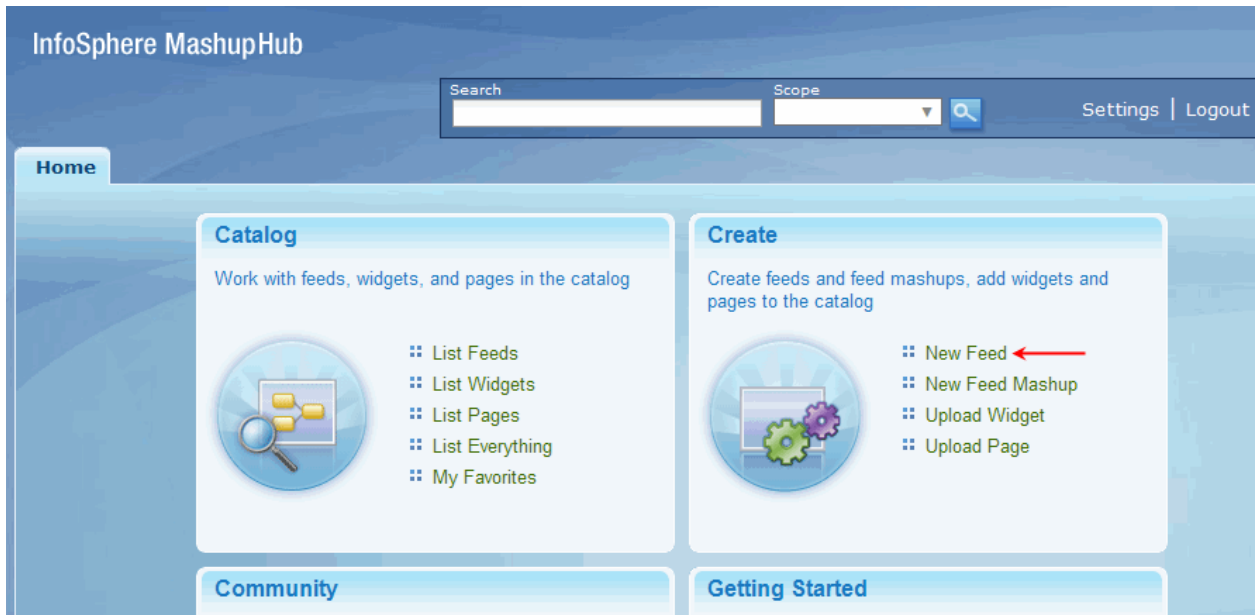


Figure 19. Selecting New Feed

What is a Feed?

Data feed is a mechanism for users to receive updated data from data sources (IMS is the data source for the feed in this tutorial). Feeds are commonly used by real-time applications as well as on the world-wide web. The latter are also called **Web feeds**.

4. Select **IMS Transaction** and click **Next**.

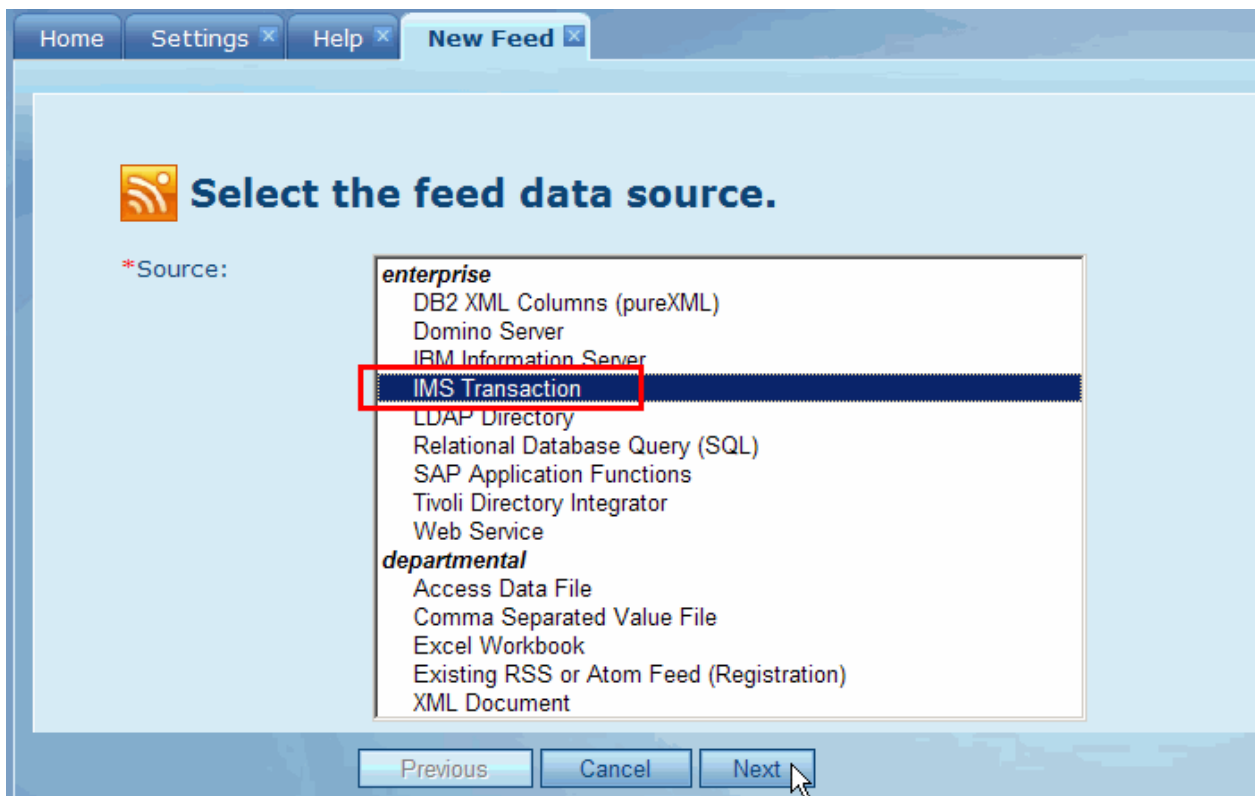


Figure 20. Selecting IMS Transaction as the feed data source

A new tab with a randomly generated feed number displays, as shown in the following figure. Note that the feed number you see is likely going to be different from the one shown in the screen capture.

5. Specify **Host Name**, **Port number**, **Data Store Name** and **XML File** as shown in the following figure. Use the following values:

Host Name or IP Address: ZSERVEROS.DFW.IBM.COM

Port: 9999

Data Store Name: IMSC

The **XML File** (correlator file) generated from your IMS Web 2.0 project can be found at the following location:

C:\Workspaces7.5\SANDBOX\SampleFeedApp\Generation\Targets\IMSPHBW.xml



Figure 21. Specifying Feed properties

6. Do not specify **Security Credentials** for this tutorial. Click **Next**.
7. Next, specify the information to expose in this feed, and provide default values for these parameters, as shown in the following figure.

Parameter Name	Hide	Default Value
in_ll	<input checked="" type="checkbox"/>	
in_zz	<input checked="" type="checkbox"/>	
in_trcd	<input checked="" type="checkbox"/>	IVTNO
in_cmd	<input type="checkbox"/>	DISPLAY
in_name1	<input type="checkbox"/>	SMITH
in_name2	<input type="checkbox"/>	
in_extn	<input type="checkbox"/>	
in_zip	<input type="checkbox"/>	

Figure 22. Specifying Feed parameters

Parameters that you specify to hide in this page will not be exposed to feed users. In this tutorial:

- We want to hide the LL and ZZ information in the input message from our users.
- We want to hide the transaction code from feed users, but we do want to specify the transaction code for this feed. Otherwise, the feed will not work.
- We are allowing feed users to specify what information (first name, lastname, phone extension, or zip code) they want to look up (DISPLAY is the default value unless the users specify otherwise).

8. Finally, provide a title and description for this feed.

The screenshot shows a web interface for creating a feed. At the top, there are tabs for 'Home' and 'Feed 31'. The main heading is 'Specify the following information'. The form contains the following fields and options:

- Title:** A text input field containing 'Sample Feed App'.
- Description:** A text input field containing 'Sample Feed'.
- Version:** A text input field containing '1.0'.
- Tags:** An empty text input field.
- Permissions:** Three radio button options: 'Public (All users can view the feed)', 'Private (Feed is invisible to all other users)', and 'Custom (Custom permission settings)'. The 'Public' option is selected.
- Advanced:** A list of expandable sections: 'Categories', 'Related entries already in the catalog', 'Technical Documentation', and 'Caching data', each with a play button icon.

At the bottom of the form, there are three buttons: 'Previous', 'Cancel', and 'Finish'. A mouse cursor is pointing at the 'Finish' button.

Figure 23. Specifying Feed title and description

9. The feed is created. Click **View feed** to test it, as shown in the following figure.


The screenshot shows a confirmation page with the following content:

- Feed successfully saved**
- The feed named Sample Feed App has been saved in the catalog.
- What's Next?**
- A list of actions, each preceded by a double colon icon (>::):
 - [View feed](#) (highlighted with a red arrow)
 - [View feed details](#)
 - [Edit feed details](#)
 - [Edit feed source](#)
 - [Create another feed](#)
 - [Close Tab](#)

Figure 24. Feed successfully created

10. Customize the parameters for your request, and click **Submit**, as shown in the following figure.

Home Sample Feed App Parameters

 **Please enter parameters for this feed.**

in_extn:

in_cmd:

in_name1:

in_name2:

in_zip:

Figure 25. Testing the feed

11. Take a look at the source of the feed to verify that the correct information is retrieved (You can right click in the browser window and select **View Source**, as shown in the following two screen captures.)

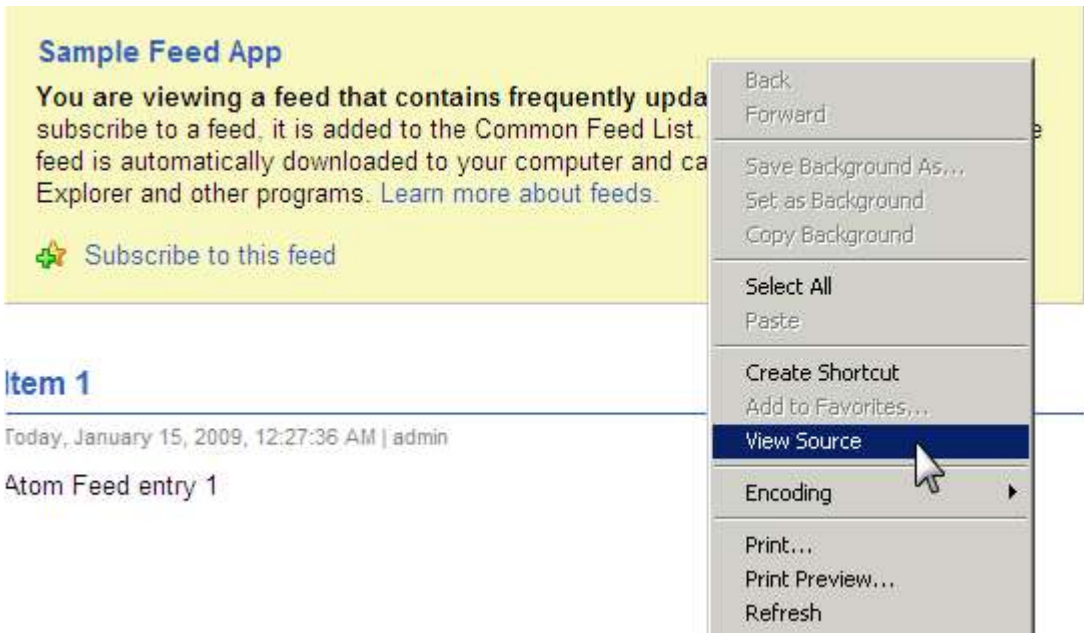


Figure 26. Selecting “View Source” to show the information returned by the feed

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <id>https://localhost:9444/mashuphub/client/plugin/generate/entryid/30/pluginid/7</id>
  <subtitle type="text">Sample Feed</subtitle>
  <link href="https://localhost:9444/mashuphub/client/plugin/generate/entryid/30/pluginid/7">
  <updated>2009-01-19T17:29:32.578Z</updated>
  <title type="text">Sample Feed App</title>
  <generator>InfoSphere MashupHub</generator>

  <entry xmlns="http://www.w3.org/2005/Atom">
    <title type="text">Item 1</title>
    <id>urn:uuid:1</id>
    <updated>2009-01-19T17:29:32.578Z</updated>
    <author>
      <name>admin</name>
    </author>
    <summary type="text">Atom Feed entry 1</summary>
    <content type="application/xml">
      <RepeatingElement xmlns="http://www.ibm.com/xmlns/atom/content/imsrecord/1.0">
        <ResponseMessage>
          <out_ll>93</out_ll>
          <out_zz>768</out_zz>
          <out_msg>ENTRY WAS DISPLAYED</out_msg>
          <out_cmd>DISPLAY</out_cmd>
          <out_name1>SMITH</out_name1>
          <out_name2>JOHN</out_name2>
          <out_extn>5639</out_extn>
          <out_zip>91234</out_zip>
          <out_segno>0001</out_segno>
        </ResponseMessage>
      </RepeatingElement>
    </content>
  </entry>
</feed>
```

Figure 27. Correct information returned by the feed

Congratulations! You have successfully created a feed in InfoSphere MashupHub that accesses the phonebook application and retrieves the phonebook entry for SMITH.

XML format for a feed

Each feed has an outmost element of <feed>.

Each feed has:

- An <id> element that contains the URL in the MashupHub catalog that is assigned to the feed.
- A <subtitle> element that contains the description of the feed that is specified when the feed is created.
- A <title> element that contains the title of the feed that is specified when the feed is created.
- An <entry> element that contains the content that a feed viewer displays.
- The title is always Item 1 for an IMS feed.
- The <content> element contains the actual data from the feed. Inside the <content> element:
 - There is only one occurrence of the <RepeatingElement> element.
 - The repeating element contains the <ResponseMessage> element that holds the data.

Task 2. Create a mashup

Objective

Use **IBM Lotus Mashups** to create a simple web application (also referred to as a “**mashup**”) from the IMS phone book feed and other components. The application should enable users to look up a person’s record in the phone book by entering his or her last name.

What is a mashup?

A **mashup** is a web application that combines data from more than one source into a single integrated tool. The term **mashup** implies easy, fast integration of widgets, feeds and other data sources to produce results that were not the original goal of the data owners. An example is the use of cartographic data from Google Maps to add location information to real-estate data, thereby creating a new and distinct web service that was not originally provided by either source.

The following figure shows what we would like to create to utilize the IMS Phonebook feed. We would like to allow the users to enter the last name of the person to look up in the phonebook, and the mashup application will return the information for that person. When users click the zip code, and application will display a map for the area for that zip code.



Figure 28. A Web mashup application that uses the IMS phone book feed

Step 1: Adding an IMS feed to Lotus Mashups

1. Start **Lotus Mashups** by selecting **Start → Programs → IBM Mashup Center → Lotus Mashups**. A login screen for the **Lotus Mashups** displays in a new tab in the browser window, as shown on the following figure.



Figure 29. Lotus Mashups login screen

2. Sign in using the following information, if you are presented with the signin page:
 - User ID: `admin`
 - Password: `password`
1. Once signed in, switch back to the **InfoSphere MashupHub** for the moment by clicking its tab in the browser.
2. If you left **InfoSphere MashupHub** exactly as it was at the end of the previous part of the tutorial, your screen should look similar to the one in Figure 25, with a tab that says “Sample Feed App”.
 - If this is the case, click the “**Sample Feed App**” tab, then click “**View feed details**” as shown in Figure 30, and proceed to Step 7.
 - If the “**Sample Feed App**” tab is missing for any reason (for example if you logged out of **InfoSphere MashupHub**), Steps 5 and 6 describe how to bring it back.

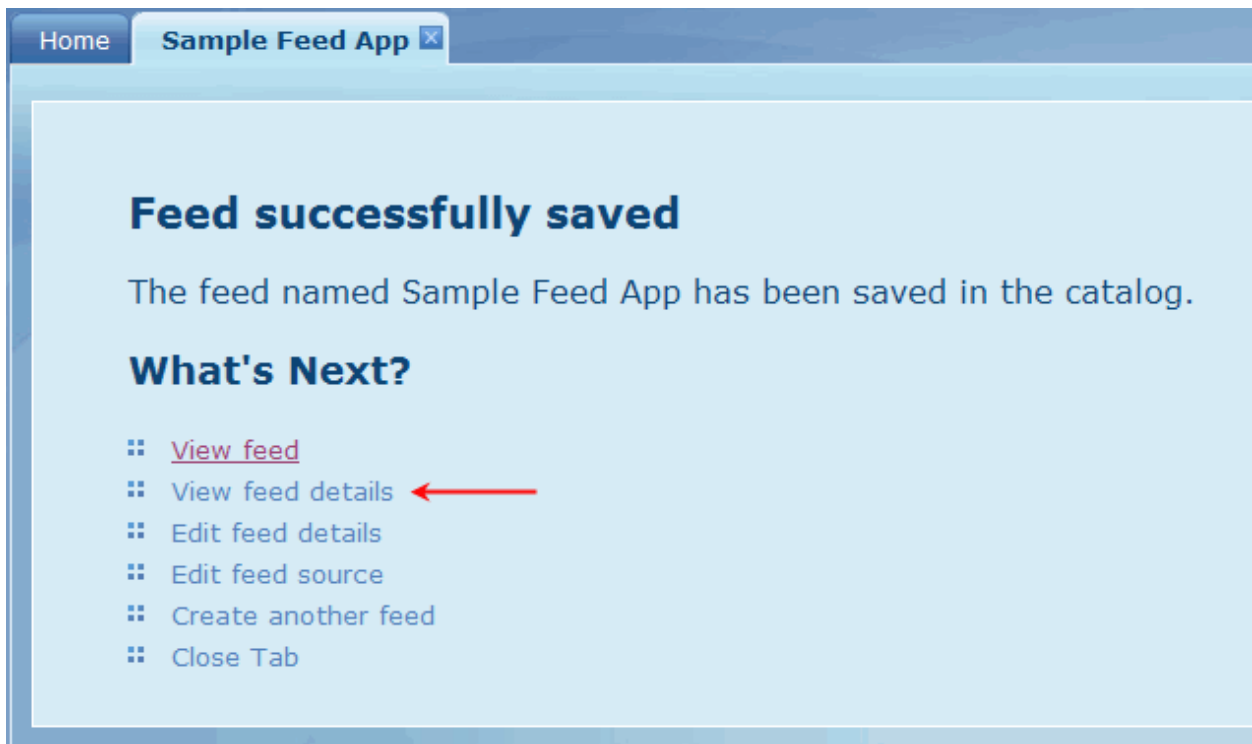


Figure 30. Selecting “View feed details”

3. On the “Home” tab, click “List Feeds” as shown in the following Figure.

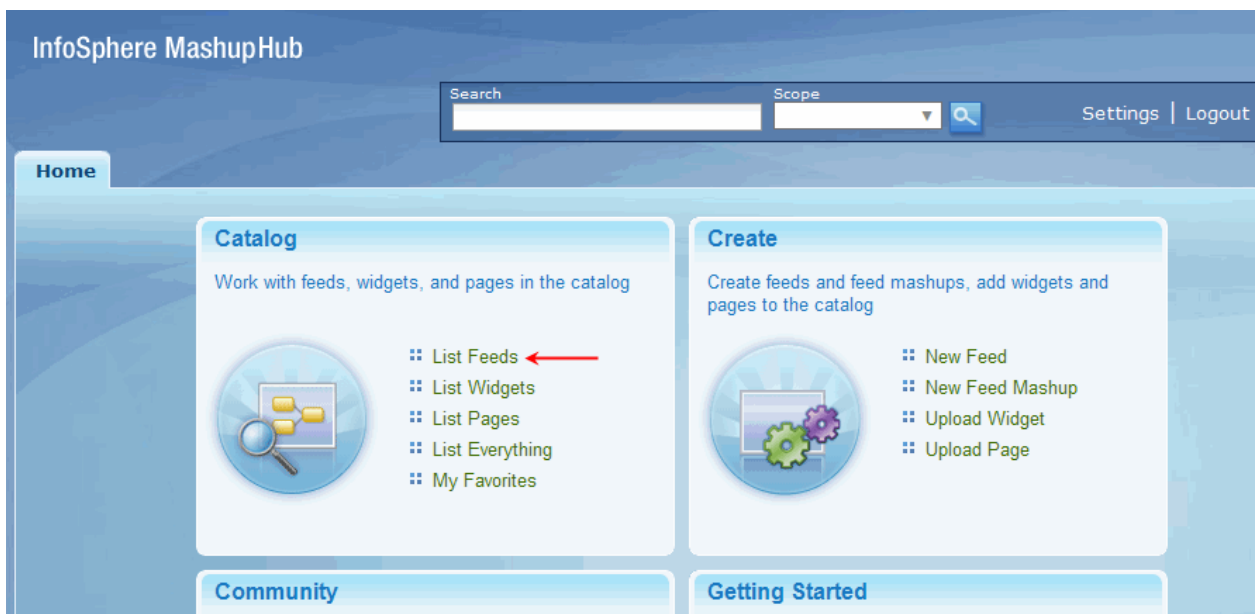


Figure 31. InfoSphere MashupHub home screen

Find the “**Sample Feed App**” entry in the **Title** column (if it doesn’t show up on the first page, you might need to click the blue arrow to go to the next page, as shown in the following Figure) . Select the “**Sample Feed App**” to open **Feed Details** screen.

Title	Tags	Updated
Aggregate Policies Mashup	aggregate functions, transform operator, example mashup	12/10/08 3:38 PM
CD Warehouse	cd, information, xml	12/10/08 3:38 PM
CD Warehouse Mashup	group operator, xml, example mashup	12/10/08 3:38 PM
CSV Sample 1	csv, sample	12/10/08 3:38 PM
Country GMT Service	country, gmt, time, timezone	12/10/08 3:38 PM
Country List Service	country, list	12/10/08 3:38 PM
Disease Outbreak Mashup	transform operator, merge operator, georss, tutorial	12/10/08 3:38 PM
Expiring Policies Mashup	variables, transform operator, functions, example mashup	12/10/08 3:38 PM
GPA Query	access, gpa, statistics	12/10/08 3:38 PM
IBM Emerging Tech Blog	emerging, technology, blog	12/10/08 3:38 PM
Insurance Mashup	merge operator, example mashup, weather	12/10/08 3:38 PM
MyCo Customer list	excel, customer	12/10/08 3:38 PM
National Park Forecast Mashup	for each operator, weather, example mashup	12/10/08 3:38 PM
National Park XML	national park, zipcode, xml	12/10/08 3:38 PM
New alphaWorks Technologies	alphaworks, emerging, technology	12/10/08 3:38 PM
Periodic Table Lookup Service	element, atom, chemistry	12/10/08 3:38 PM
Policy Holders	sql, relational, insurance	12/10/08 3:38 PM
Policy Holders Mashup	filter operator, example mashup	12/10/08 3:38 PM
Restaurants CSV	csv, restaurants	12/10/08 3:38 PM
Sample Feed App		1/15/09 2:48 PM

Figure 32. Selecting "View feed details"

Notice the feed URL (as shown within the dashed box in Figure 33):

[http://localhost:9080/mashuphub/client/plugin/generate/entryid/30/
pluginid/7?in_cmd=DISPLAY&in_name1=SMITH&in_name2=&in_extn=&in_zip=](http://localhost:9080/mashuphub/client/plugin/generate/entryid/30/pluginid/7?in_cmd=DISPLAY&in_name1=SMITH&in_name2=&in_extn=&in_zip=)

Clicking or manually entering the URL is another way of invoking an IMS feed. Notice that feed parameters (such as “DISPLAY” and “SMITH”) are contained within the feed URL.

4. Select **Add to Lotus Mashups** from the list of **Actions**.

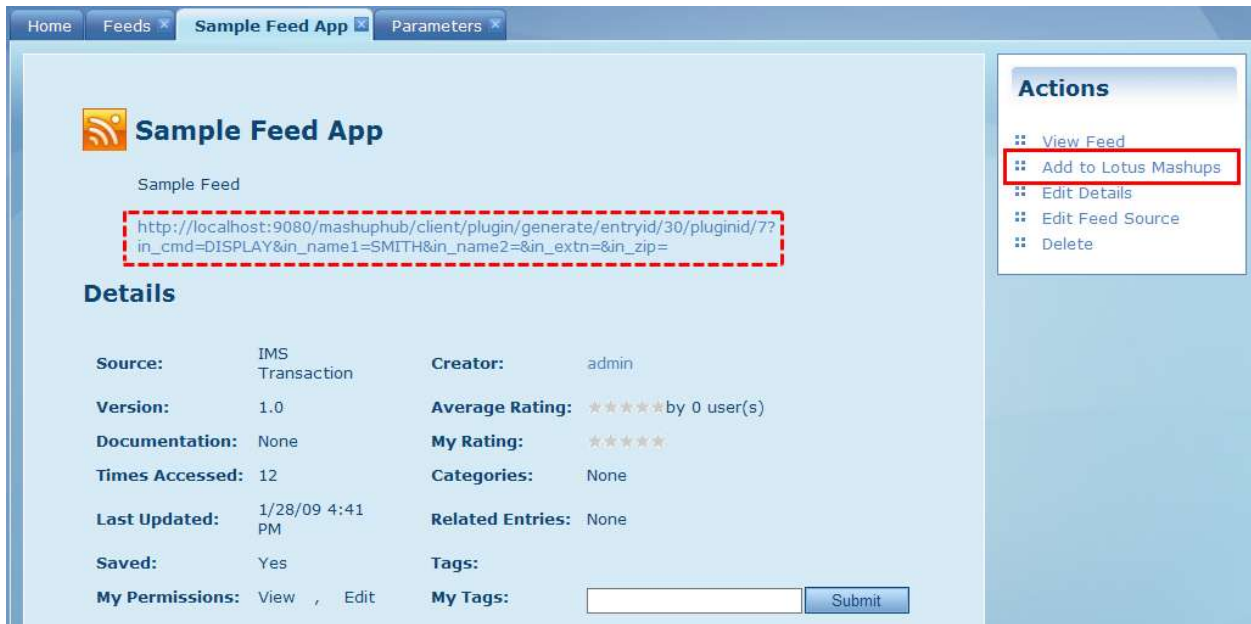


Figure 33. Selecting “Add to Lotus Mashups”

5. Do not modify any options on the following screen; just click **Next**.



Figure 34. Lotus Mashups server info

6. On the following screen (click **Yes** if security warning pops up), specify a feed title and the category this feed should belong to in Lotus Mashups. For this demo,
- For the title, type Phonebook Feed.
 - Set the category to Demo.
 - Click **Finish**.

The screenshot shows the 'Add to Lotus Mashups' screen. The form includes the following fields and values:

- Title:** Phonebook Feed
- Description:** Sample Feed
- Feed Url:** http://localhost:9080/mashuphub/client/plugin/generate/
- Parameters:**
 - in_cmd: DISPLAY (Type:)
 - in_name1: SMITH (Type:)
 - in_name2: (Type:)
 - in_extn: (Type:)
 - in_zip: (Type:)
- Category:** Demo
- Feed Viewer:** Data Viewer (Change Widget)
- Icon:** Change Icon (Preview)

At the bottom of the screen, there are three buttons: Previous, Cancel, and Finish. The Finish button is highlighted with a mouse cursor.

Figure 35. “Add to Lotus Mashups” screen

Notice the “Data Viewer” option for the “Feed Viewer” (Figure 35). When we added the **Phonebook Feed** to **Lotus Mashups**, we did not in fact create a brand new widget. Instead, we used an existing generic widget called **Data Viewer** and specified the URL of the Phonebook Feed as a parameter to it (“Feed URL” box on Figure 35). This notion will become important in the last optional task of this tutorial where we will create a different version of the **Phonebook Feed**. It will not be required to add this new feed to **Lotus Mashups** like we just did, but instead it will be sufficient to replace the URL of the existing **Data Viewer** widget (see Task 3, Step 36).

You will be informed that the feed has been added to Lotus Mashups. Also notice the message asking to refresh **Lotus Mashups**. Click “**View feed details**” to go back to the feed details page. It will be needed once again later on.

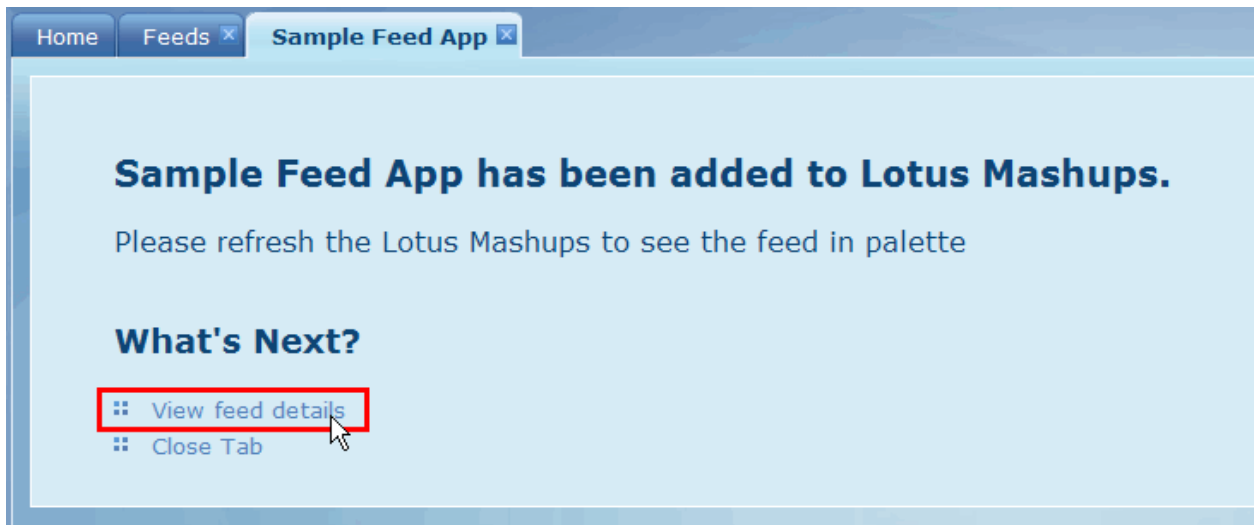


Figure 36. Feed has been successfully added to Lotus Mashups

Step 2: Adding components to a mashup page

1. Return to the IBM Lotus Mashups tab.

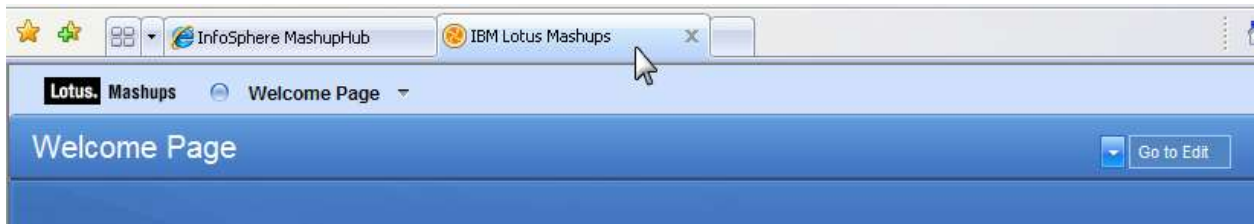


Figure 37. Back to Lotus Mashups

2. Refresh the Lotus Mashups by clicking the web browser's **Refresh** button. This is needed for the IMS feed to show up in the Lotus Mashups' palette.

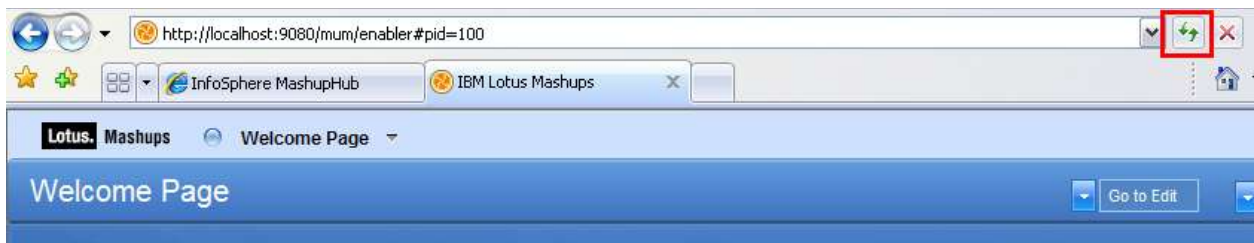


Figure 38. Refreshing Lotus Mashups

3. Mouse over Welcome Page, and select **Create a New Page** from the drop down menu.

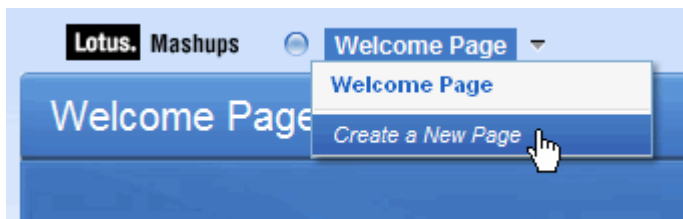


Figure 39. Creating a new page

4. Type "**Phone Book**" in the "**Create a New Page**" box and press **Enter**.
The new Phone Book page displays and is automatically set to edit mode.

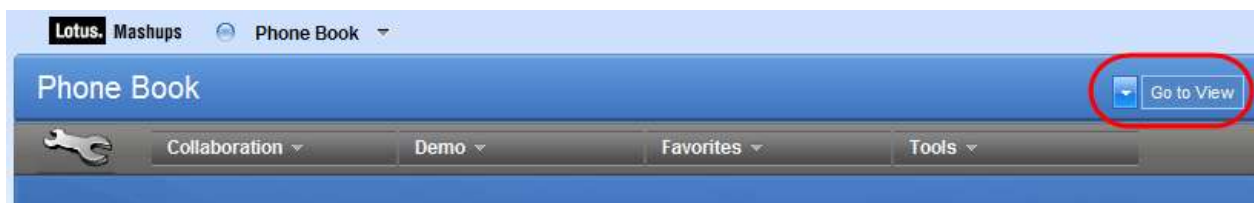


Figure 40. The Phone Book page

Lotus Mashups has two modes—edit mode and view mode. When you create a new page, you are automatically set to edit mode. You can switch back and forth between the two modes by clicking the **Go to Edit** or **Go to View** button

In edit mode, the page editor screen contains four drop down tabs – **Collaboration**, **Demo**, **Favorites** and **Tools**. Each tab contains a collection of feeds or widgets that can be mashed together to create a

Generate an IMS feed using IMS Web 2.0 Solution for InfoSphere MashupHub

© Copyright IBM Corporation 2009. All rights reserved.

dynamic web application.

5. Click the “**Favorites**” tab to expand it. Mouse over the “**User Input**” widget.



Figure 41. Selecting User Input widget

6. Click and hold the left mouse button, and then drag the **User Input** widget onto the main editor area below. Position it in the top left corner area (as shown in Figure 41) and release the mouse button.

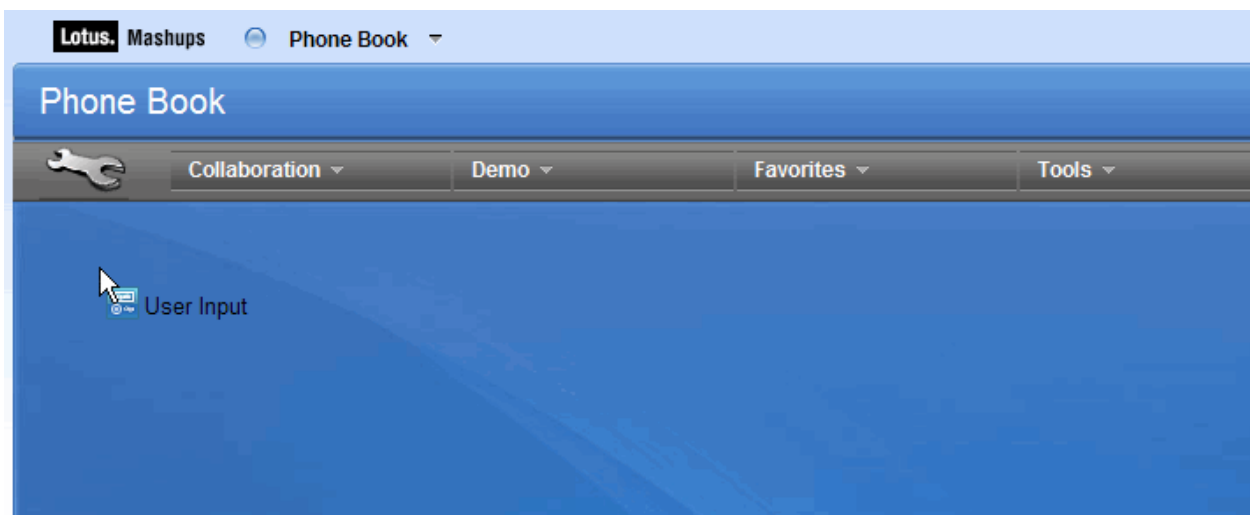


Figure 42. Dragging and dropping the User Input widget into the editor

What is a Widget?

A web widget is a portable chunk of code that can be installed and executed within any separate HTML-based web page by an end user without requiring additional compilation. They are derived from the idea of code reuse.

Lotus Mashups includes a set of widgets to provide some common features required in a Web application. The User Input widget, for example, displays one or more input fields and generates events for each input field. The event sends over the value selected or entered by the user.

Custom widgets can be created using the IWidget format and added to the MashupHub Catalog for reuse

7. **User Input** widget will appear. Reposition it as needed by dragging the gray header area.



Figure 43. User Input widget in the editor

8. Open **Settings** dialog of the **User Input** widget.

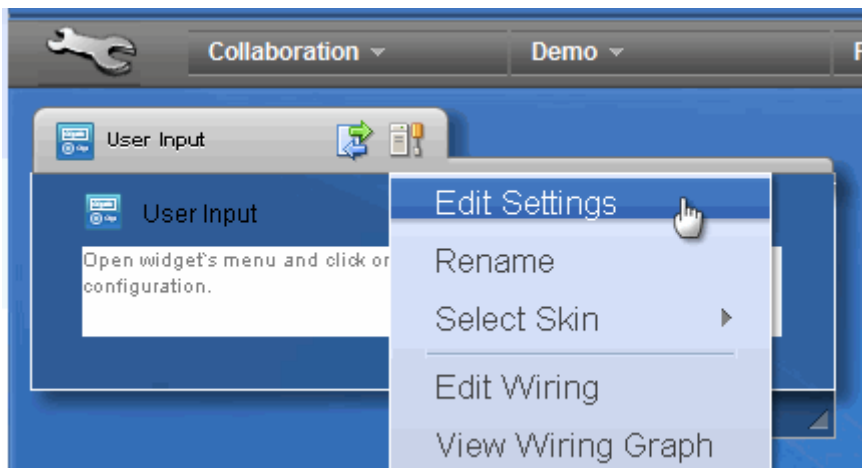


Figure 44. Selecting "Edit Settings"

9. You will now create a dialog that allows user of your application to search the phone book by entering a last name. In the **User Input Settings** dialog, click **Add** button. Then, in the “**New Field**” area, enter the following user prompt: “**Last name:**” (without quotes) and click **Save**.

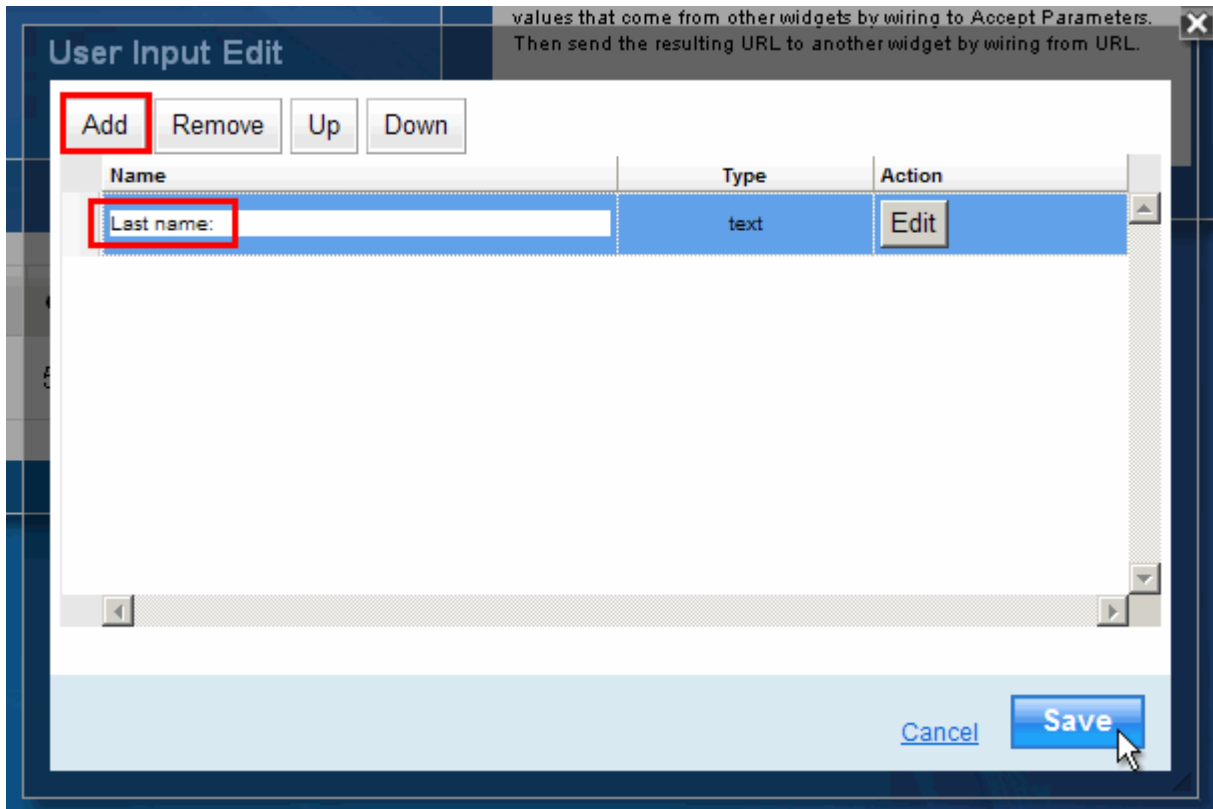


Figure 45. Creating user prompt

10. Now click the “**Demo**” tab to expand it. Mouse over the “**Phonebook Feed**”.

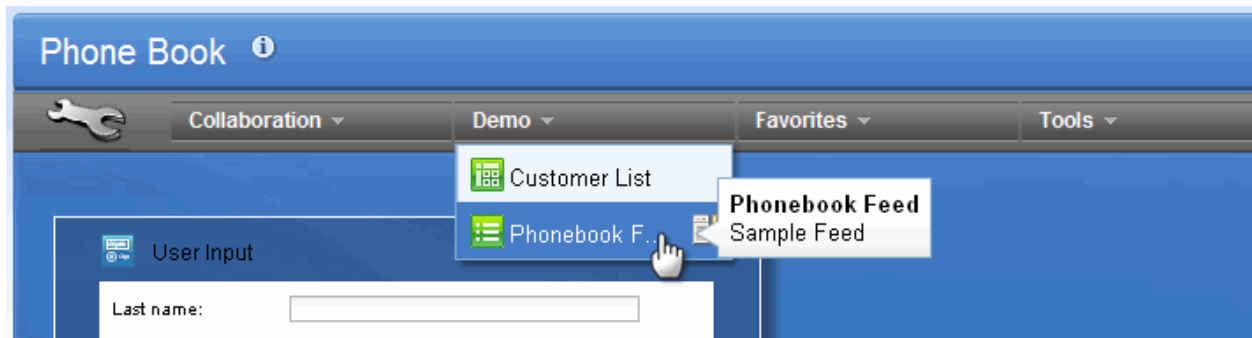


Figure 46. Phonebook Feed in the Demo tab

11. Drag and drop the **Phonebook Feed** below the **User Input** widget.



Figure 47. Dragging and dropping the Phonebook Feed into the editor

12. Click the **Options** icon located on the **Phonebook Feed** header and select “**Edit Settings**”.

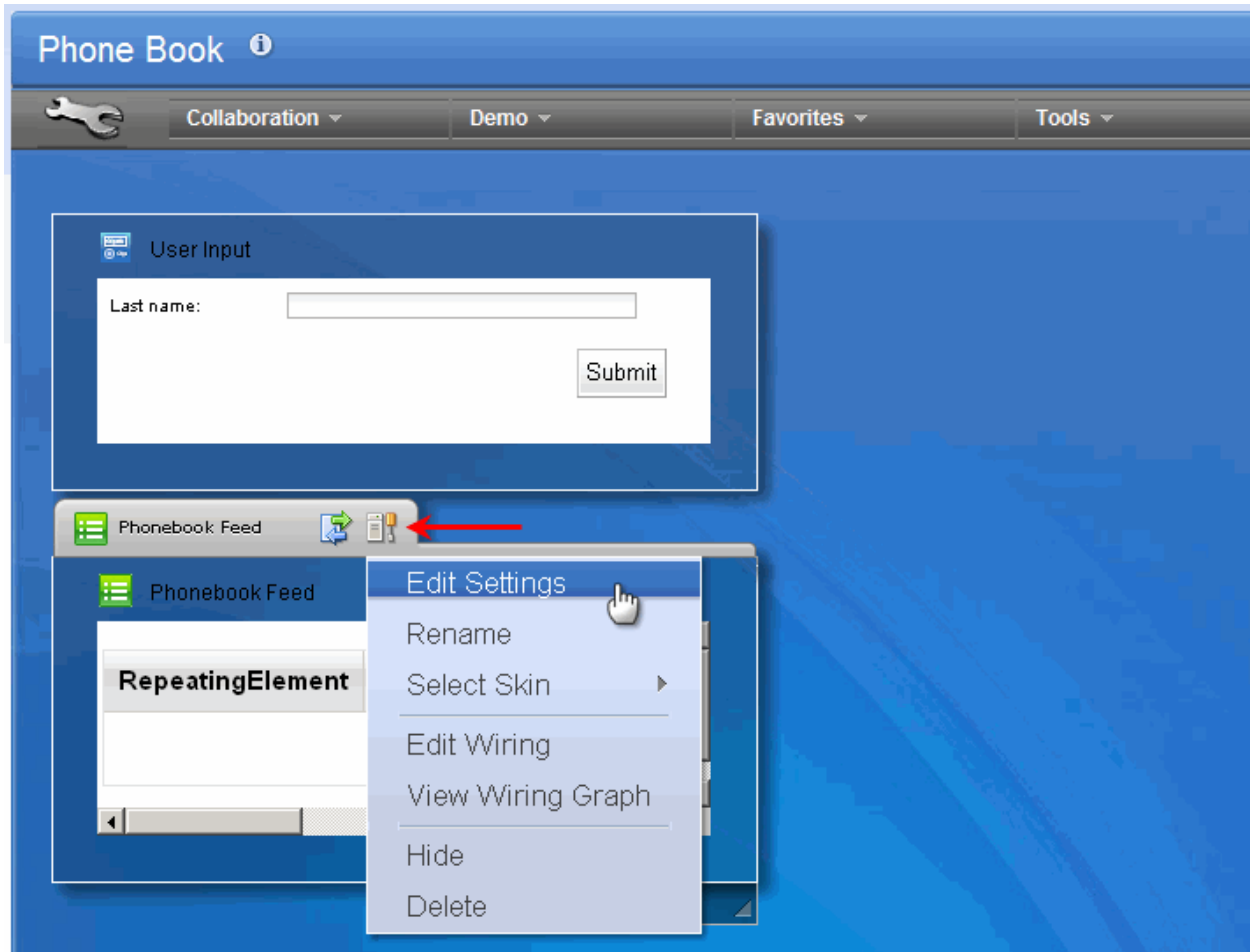


Figure 48. Selecting “Edit Settings”

13. Unselect the following columns: **RepeatingElement**, **ResponseMessage**, **out_ll**, **out_zz**, **out_msg**, **out_cmd** and **out_segno** (scroll down to see **out_segno**). This is done to prevent the mashup from displaying these utilitarian columns. Finally click **Save** to close the **Settings** dialog.

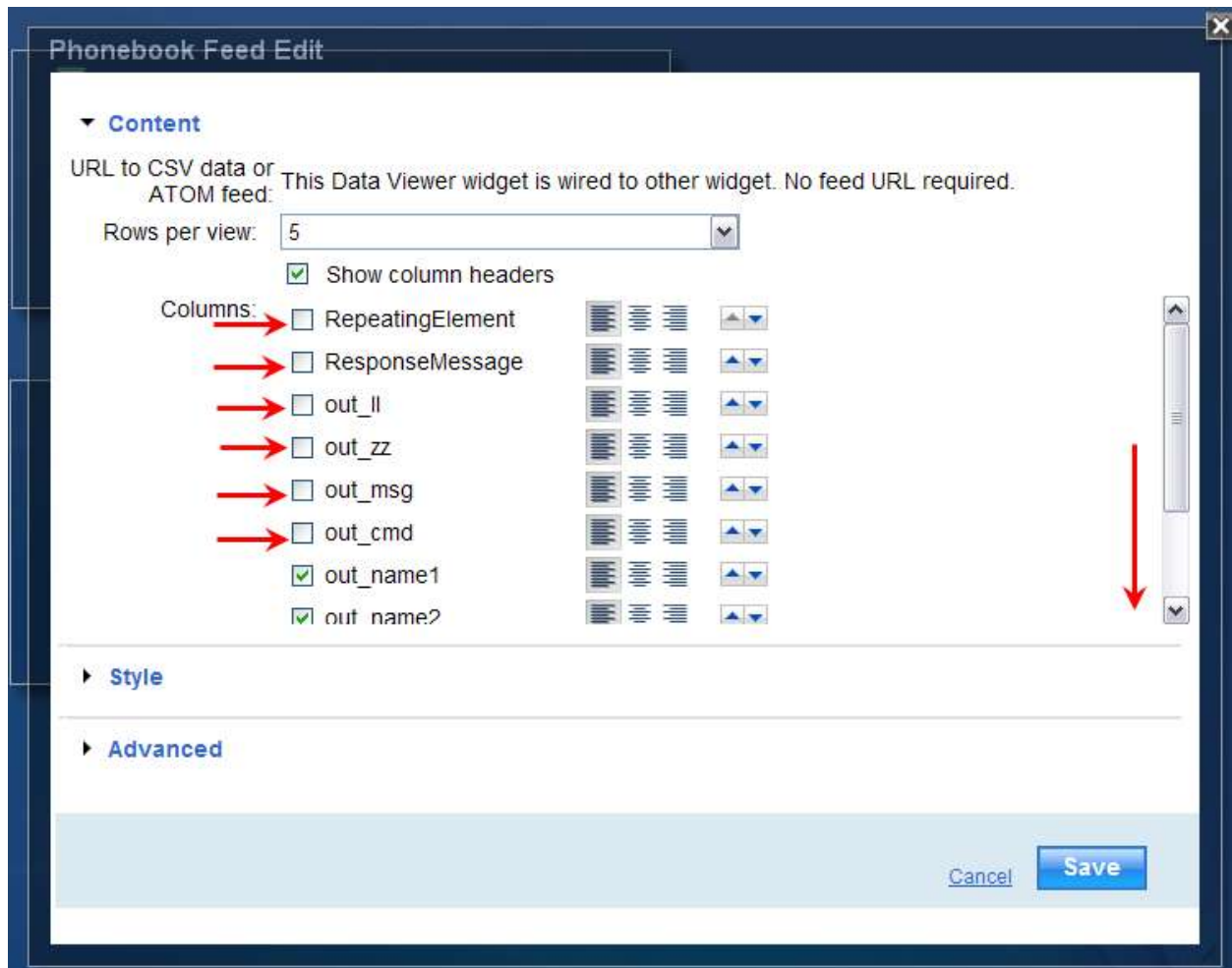


Figure 49. Hiding columns

We now have two components on the page, one to allow users to provide input (the last name for the person to look up in the phone book), and the results from the phonebook feed. However, the user input widget is not yet hooked up to query the IMS feed (Sample Feed App), nor is the widget wired with the Phonebook Feed component on the page to display the query result.

We will use the URL Customizer widget to create dynamic URLs with customized parameters.

The URL Customizer lets you specify a parameterized feed URL from the catalog or a parameterized URL on the Web, and the parameter values can come from an event that is passed from another widget on the mashup page. The URL Customizer widget automatically parses the parameters from the URL.

1. With the URL Customizer widget, we'll specify the URL of the Sample Feed App to be the base URL. The widget will load in the parameters available for the feed.
2. We'll specify that the values of the parameters should come from the User Input widget.
3. We'll specify that the parameterized URL should be sent to the Phonebook Feed.

14. Expand **Tools** tab and select the **URL Customizer** widget. Drag and drop the widget into the editor.

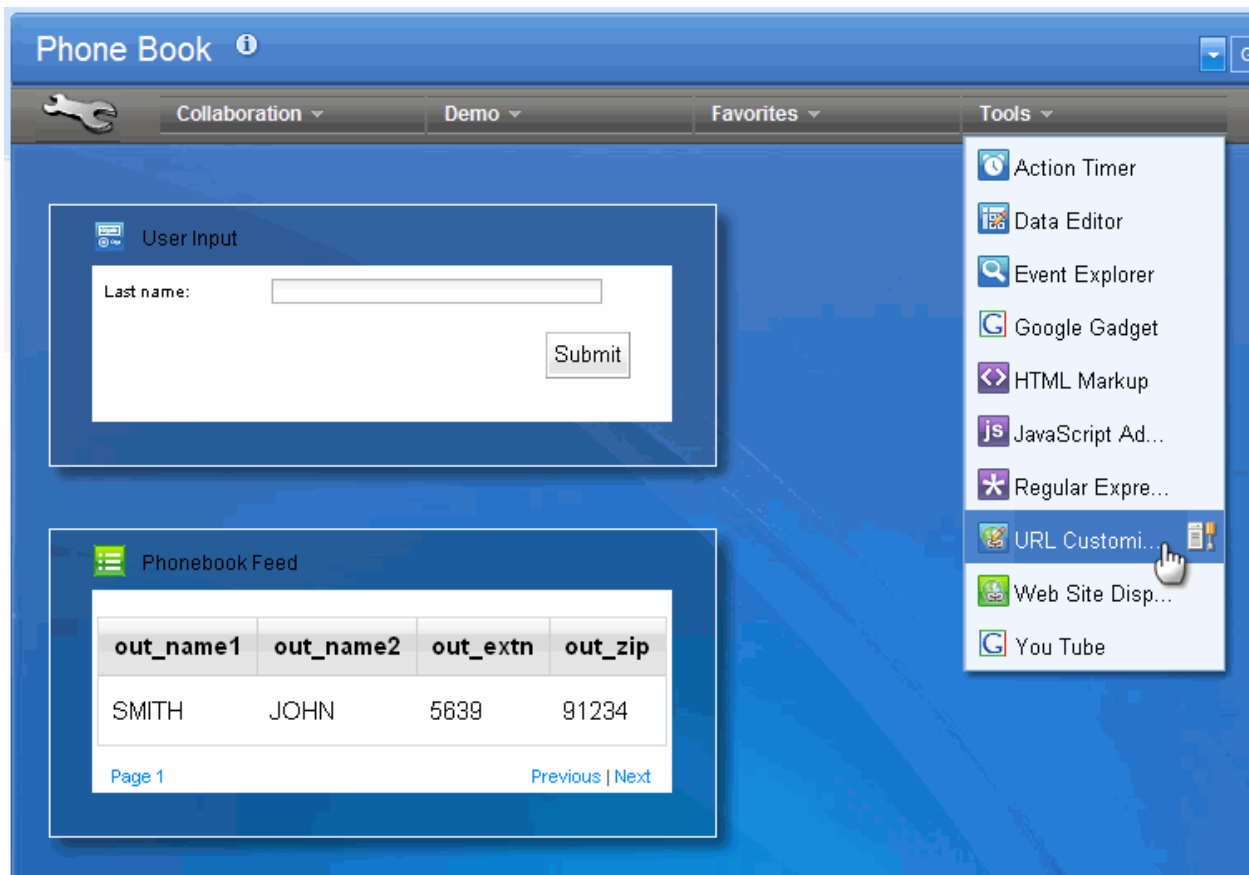


Figure 50. Selecting URL Customizer widget

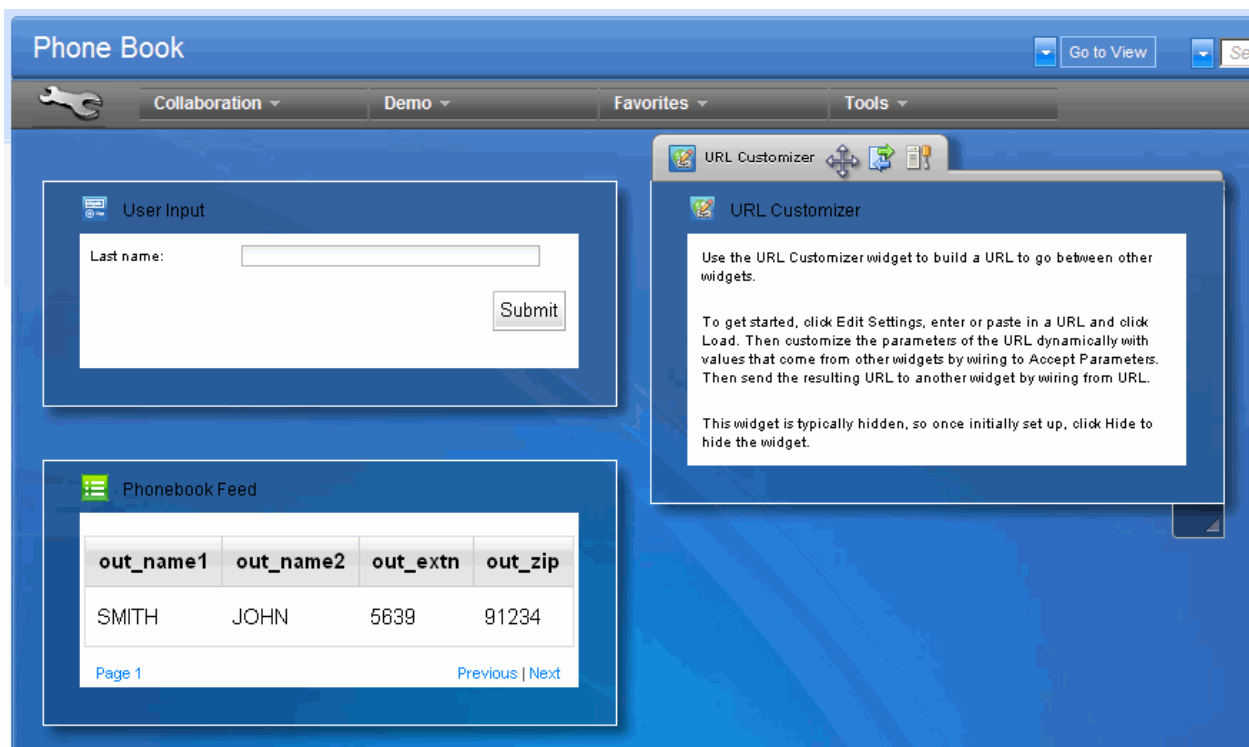


Figure 51. Placing URL Customizer into the editor area

Remember that the **IMS feed** has its input parameters such as last name contained within URL. The reason we added the **URL Customizer** widget is to be able to modify the URL of the **IMS feed** with the last name provided by the **User Input** widget.

15. Open **URL Customizer's Settings** dialog.

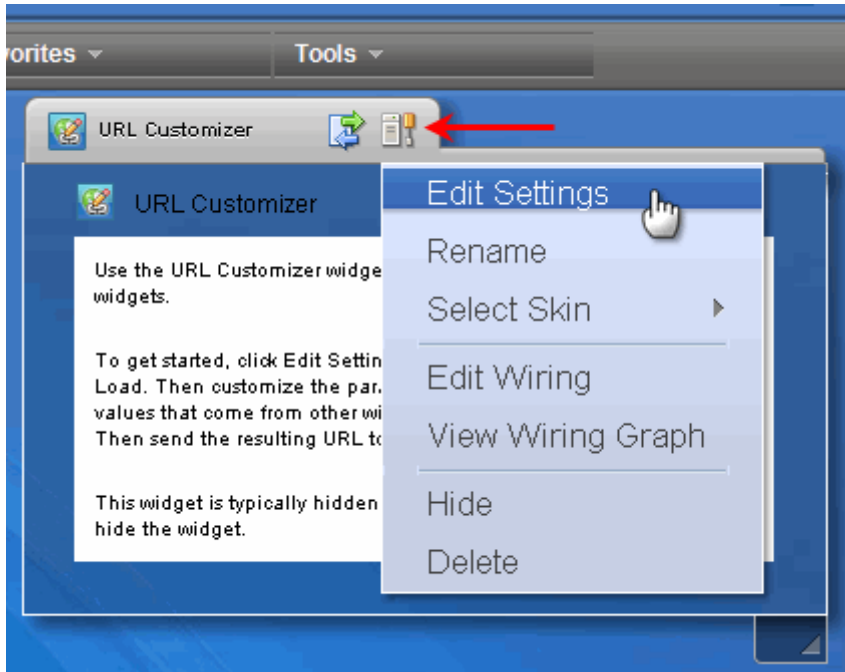


Figure 52. Opening URL Customizer's Settings dialog

16. Go to the **InfoSphere Mashuphub**, **Sample Feed App** details. Select and copy (*Ctrl-C*) the IMS feed URL.



Figure 53. Copying IMS feed's URL

- Return to the **Lotus Mashups, URL Customizer** settings. Paste (*Ctrl-V*) the URL into the text box and click **Load**.

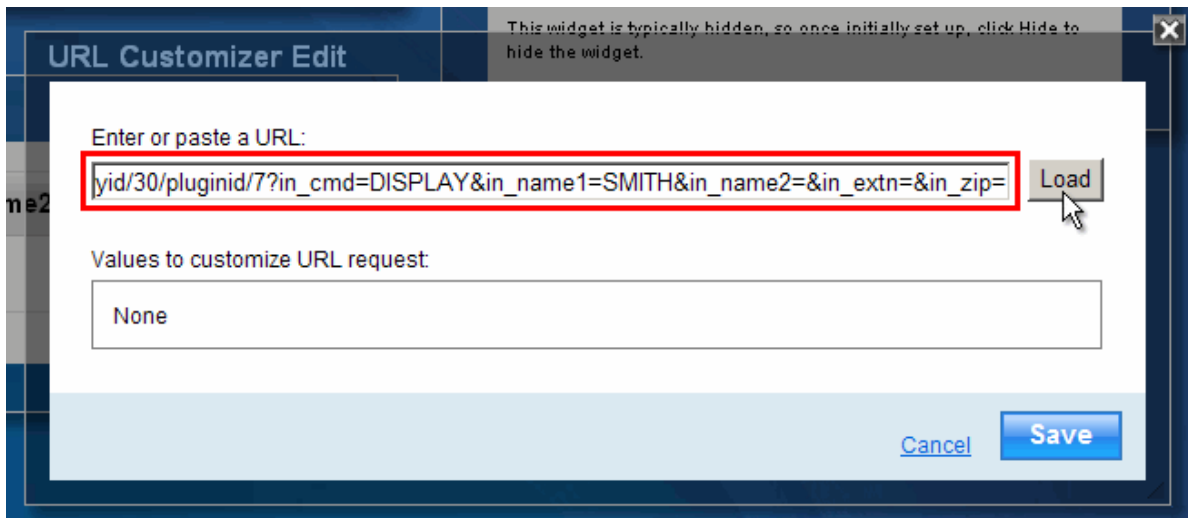


Figure 54. Pasting and loading URL of the IMS feed

- Select **in_name1** as the parameter to receive a single value (following Figure). **URL Customizer** can receive and replace a value of one of the URL parameters. We want **in_name1**, which corresponds to the last name, to be replaced. Finally, click **Save**.

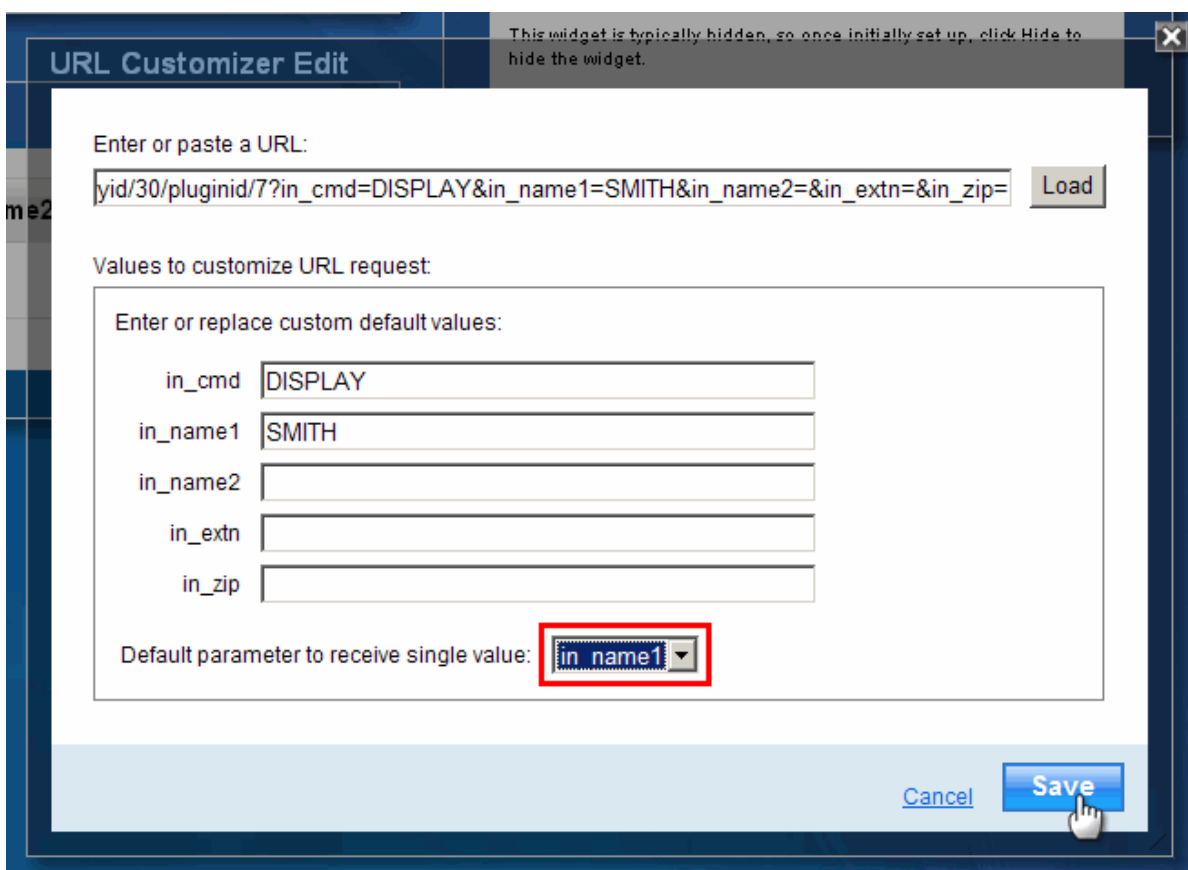


Figure 55. Changing the default parameter to receive single value

We are now going to wire the **User Input** widget with the **Phonebook** feed such that when the user enters last name and clicks **Submit** button, the **Phonebook** feed displays records of persons by that name.

Step 3: Wiring components together

Now it is time to wire all components together to make them work in tandem.

19. Mouse over the **User Input** widget and click the **Wiring** icon located to the left of the **Options** icon:

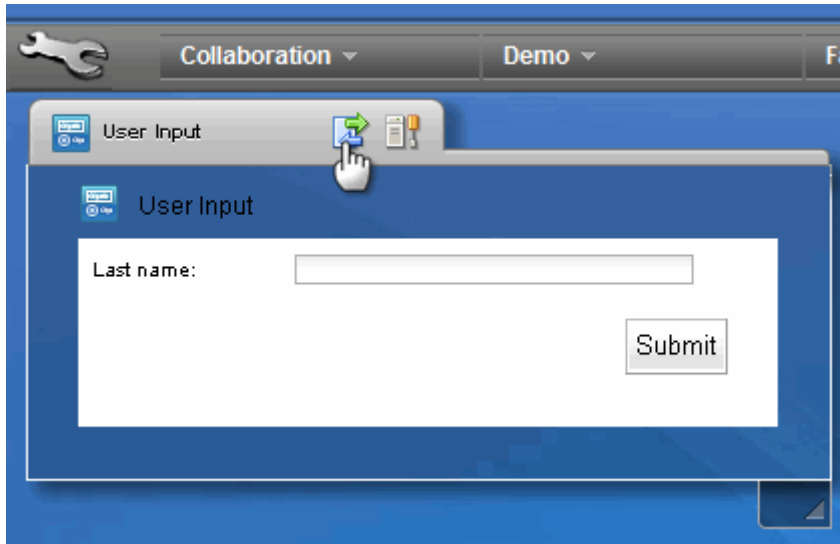


Figure 56. Wiring icon

The wiring dialog has two tabs – **Receive** and **Send**. Within each tab you can specify which data is to be received from/send to other components.

The **User Input** widget allows sending data (last name in our case) as plain text. Remember however that an **IMS feed** has input parameters such as last name contained within URL. The reason we added the **URL Customizer** widget in **Step 14** is to be able to modify the URL of the **IMS feed** with the last name provided by the **User Input** widget.

20. Within the **Send** tab of the **Wiring** dialog, select “**Last name: as Any Data**” as the content to send.

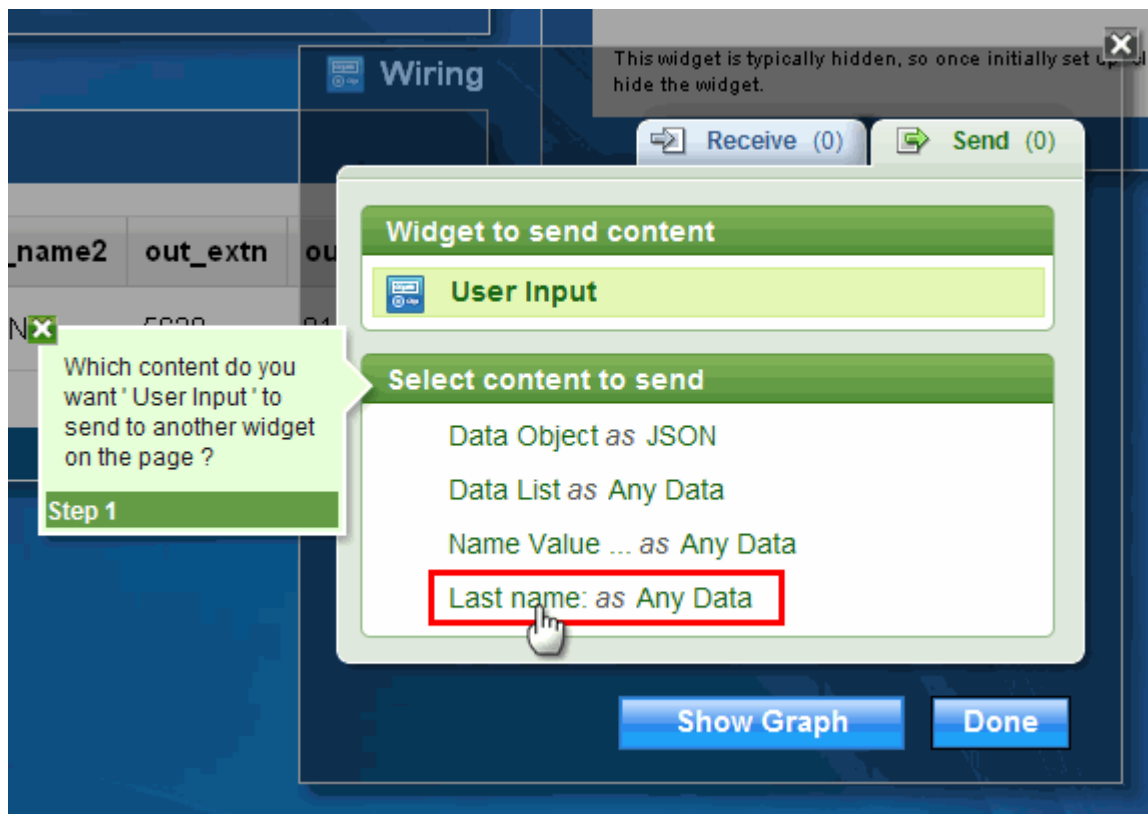


Figure 57. Selecting content to send

21. Now select **URL Customizer** as the widget to receive content. Notice how the widget gets highlighted (useful if more than one widget of the same type is used).

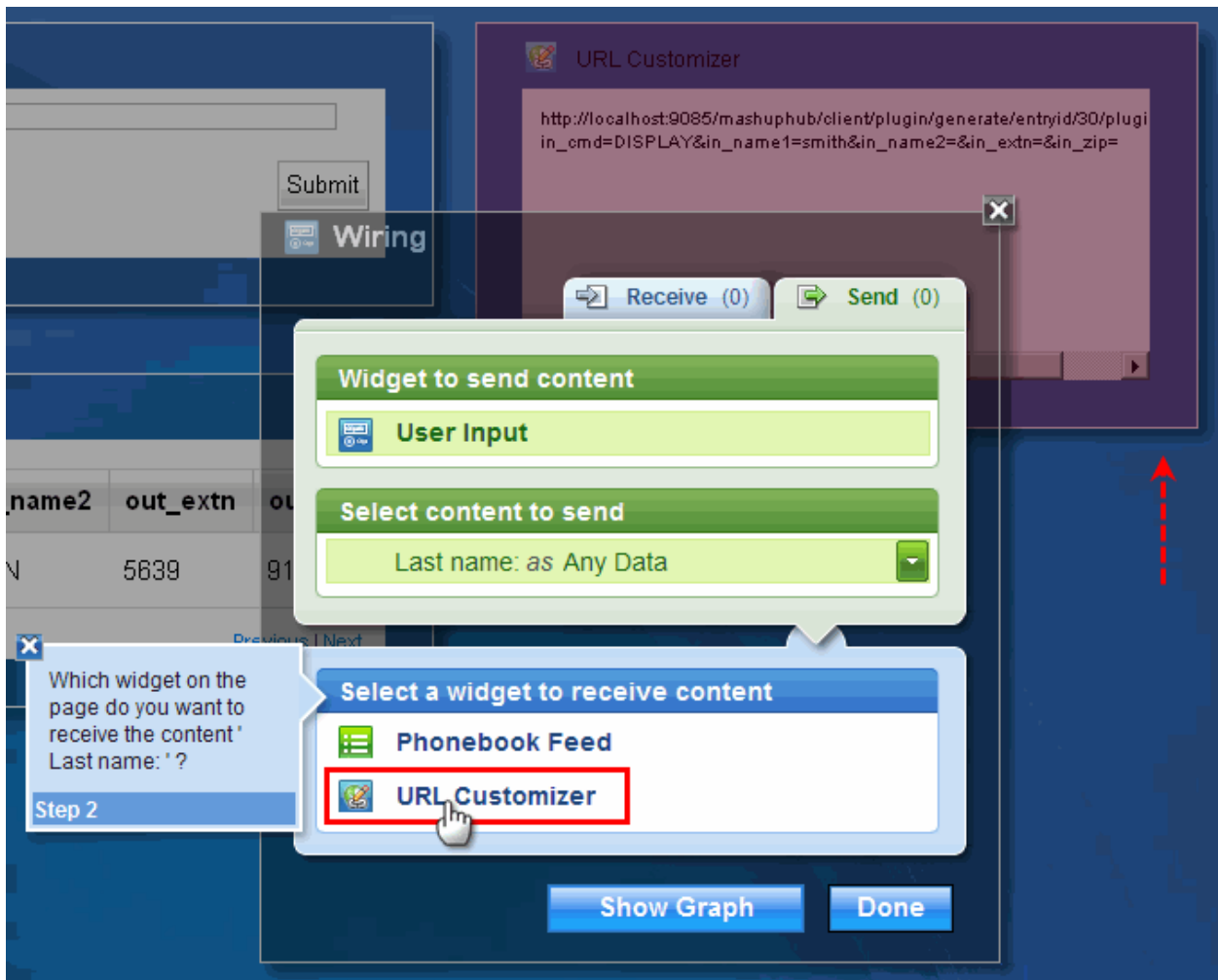


Figure 58. Selecting a widget to receive content

22. Select the check box next to “Accept Parameters *using Any Data*” and click **Done**.

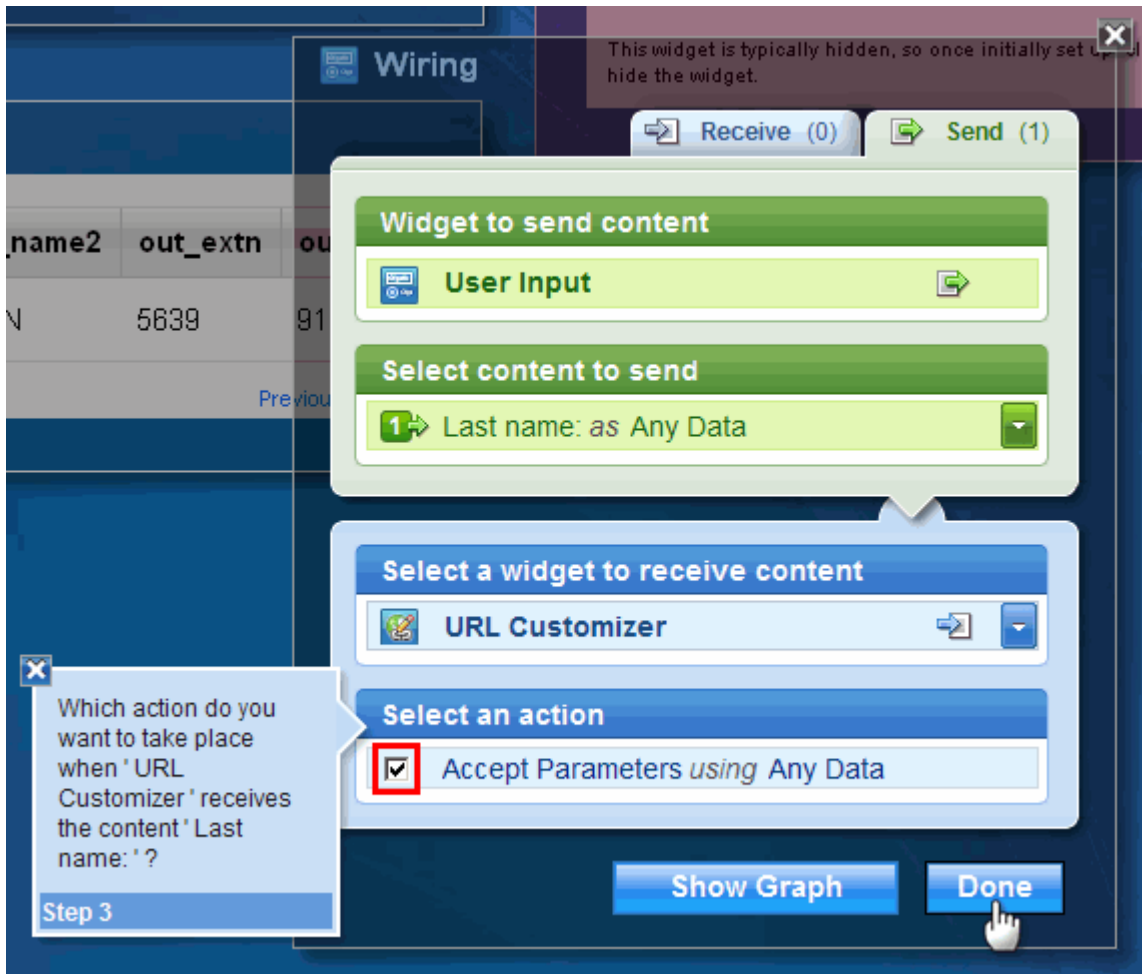


Figure 59. Selecting an action

23. Click the **Wiring** icon of the **URL Customizer** widget.

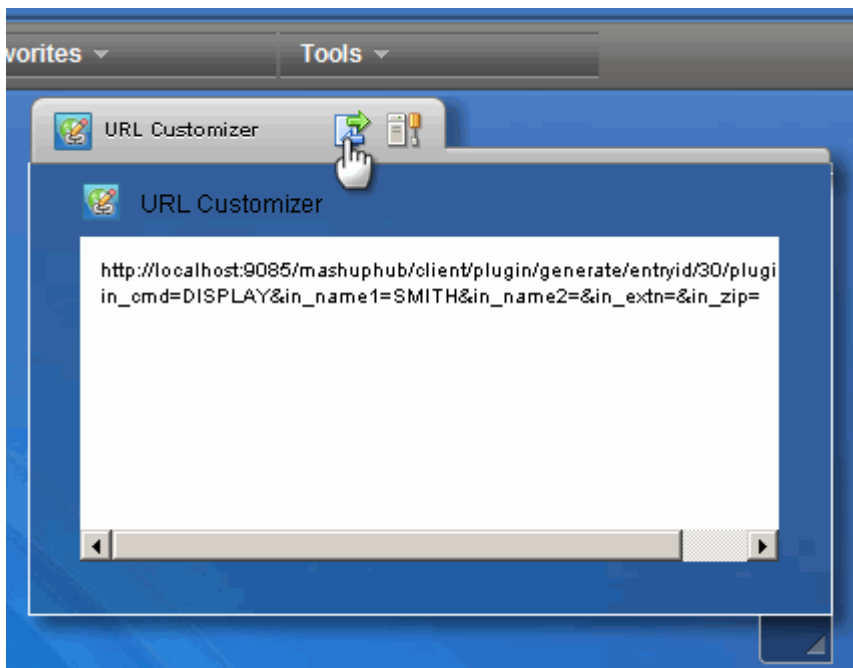


Figure 60. Wiring icon

24. Select “URL as URL” as the content to be sent.

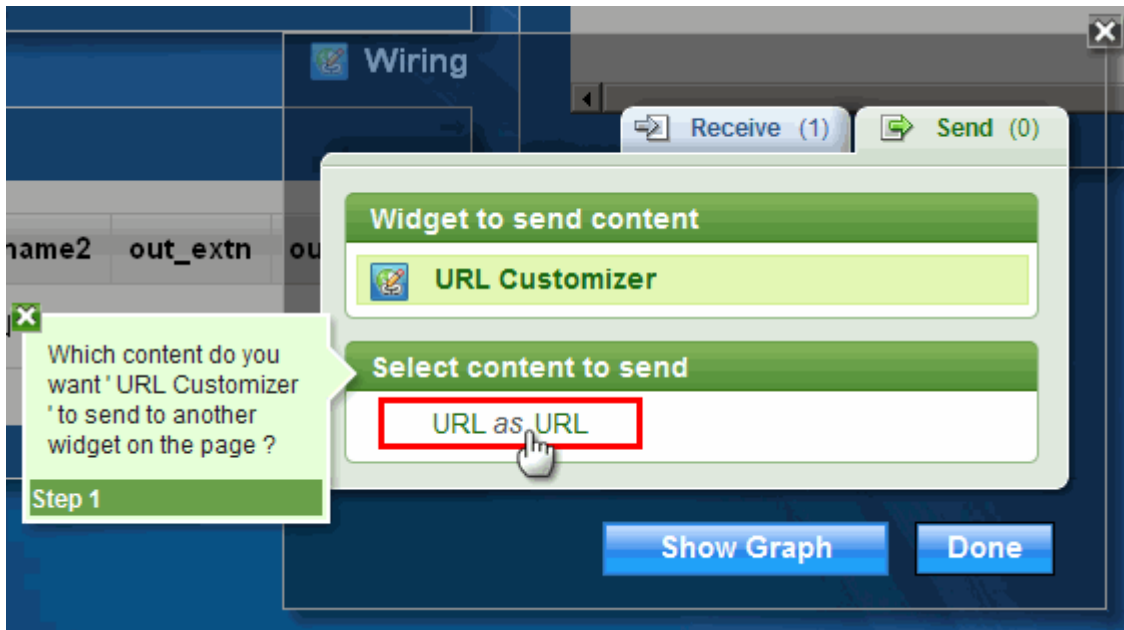


Figure 61. Selecting content to send

25. Select **Phonebook Feed** as the widget to receive content.

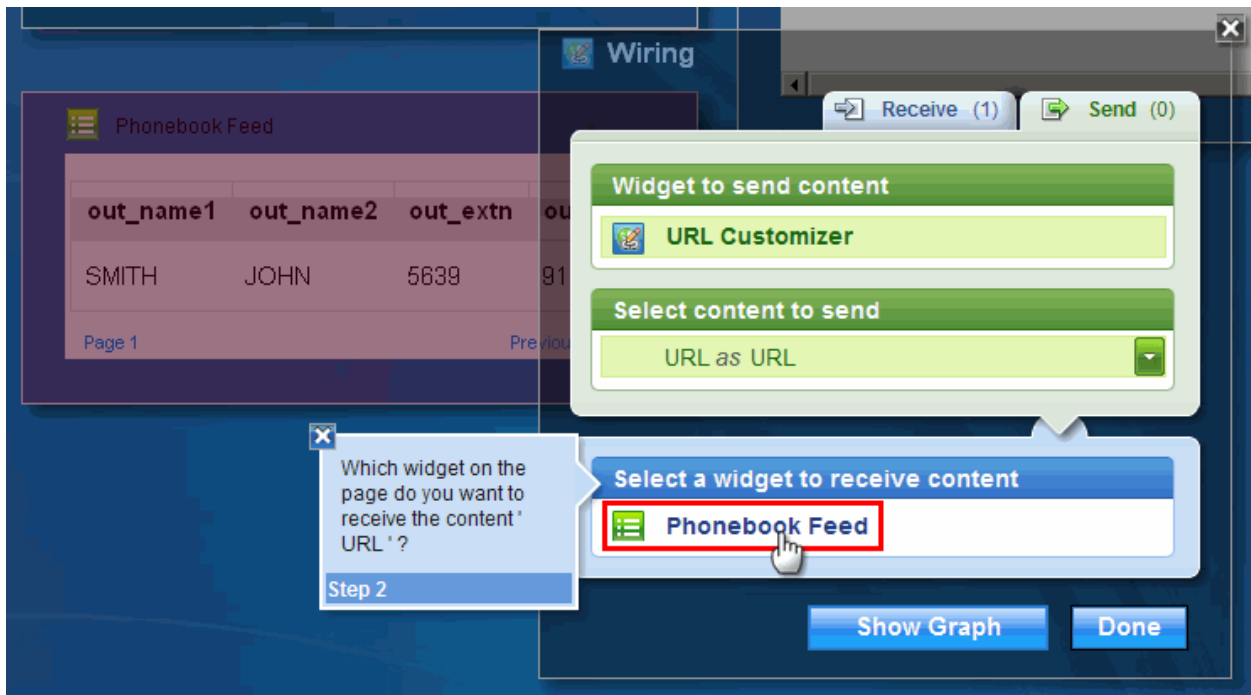


Figure 62. Selecting a widget to receive content

26. Check the selection box next to “Table using URL (ATOM)” and click **Done**.

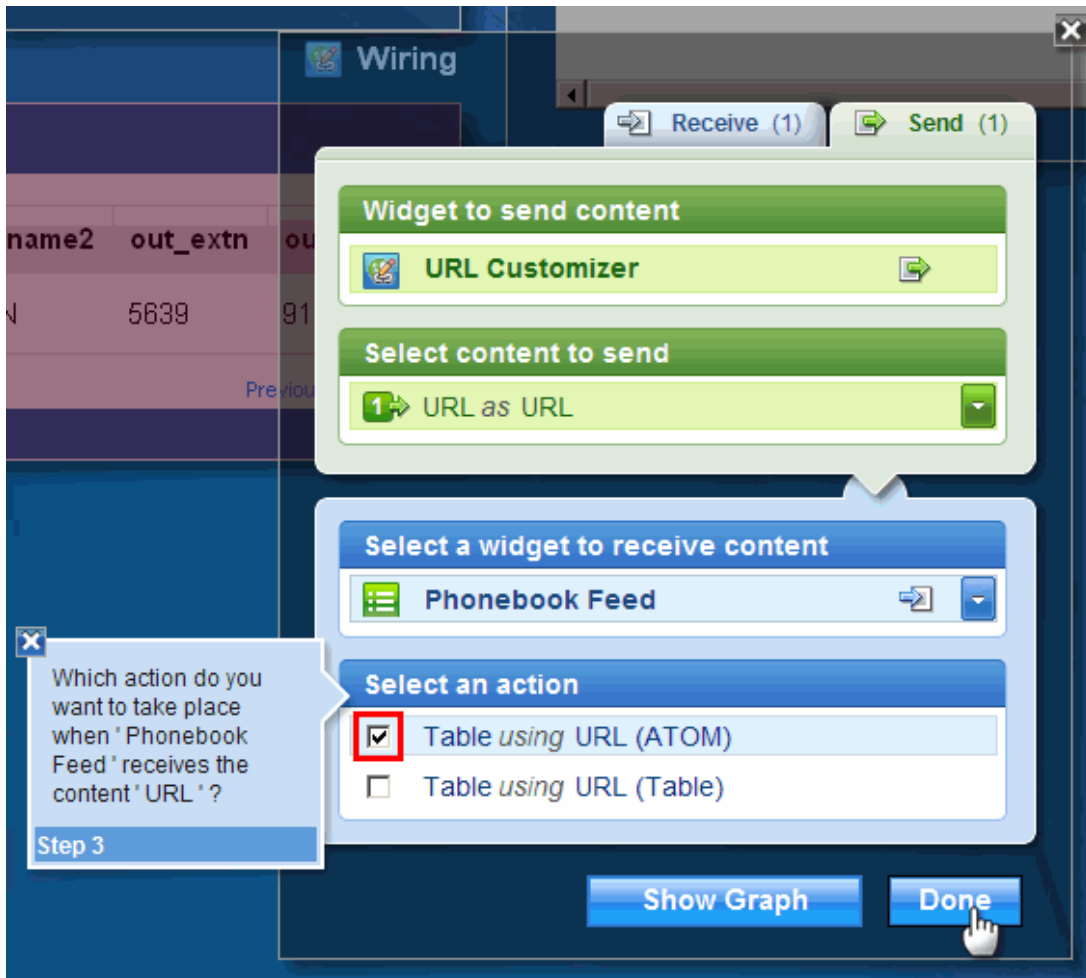


Figure 63. Selecting an action

27. Click **Hide** in the **Options** menu of the **URL Customizer** widget. This is done to hide **URL Customizer** from our Phonebook application user interface.

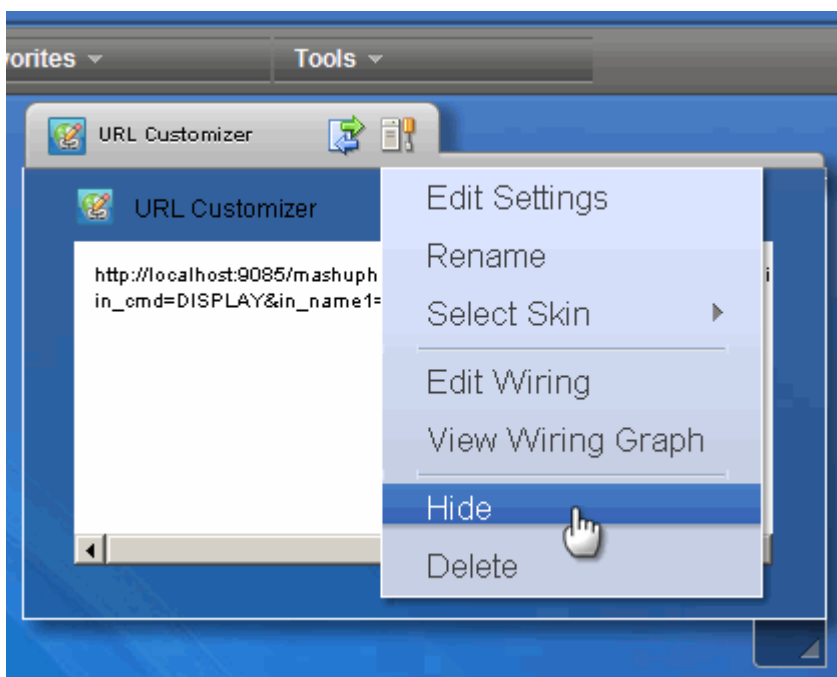


Figure 64. Hiding the URL Customizer

Once hidden, the **URL Customizer** widget disappears from the editor screen. To view or edit its settings and wiring, or to unhide it, click the “**View hidden widgets**” icon (see following Figure).

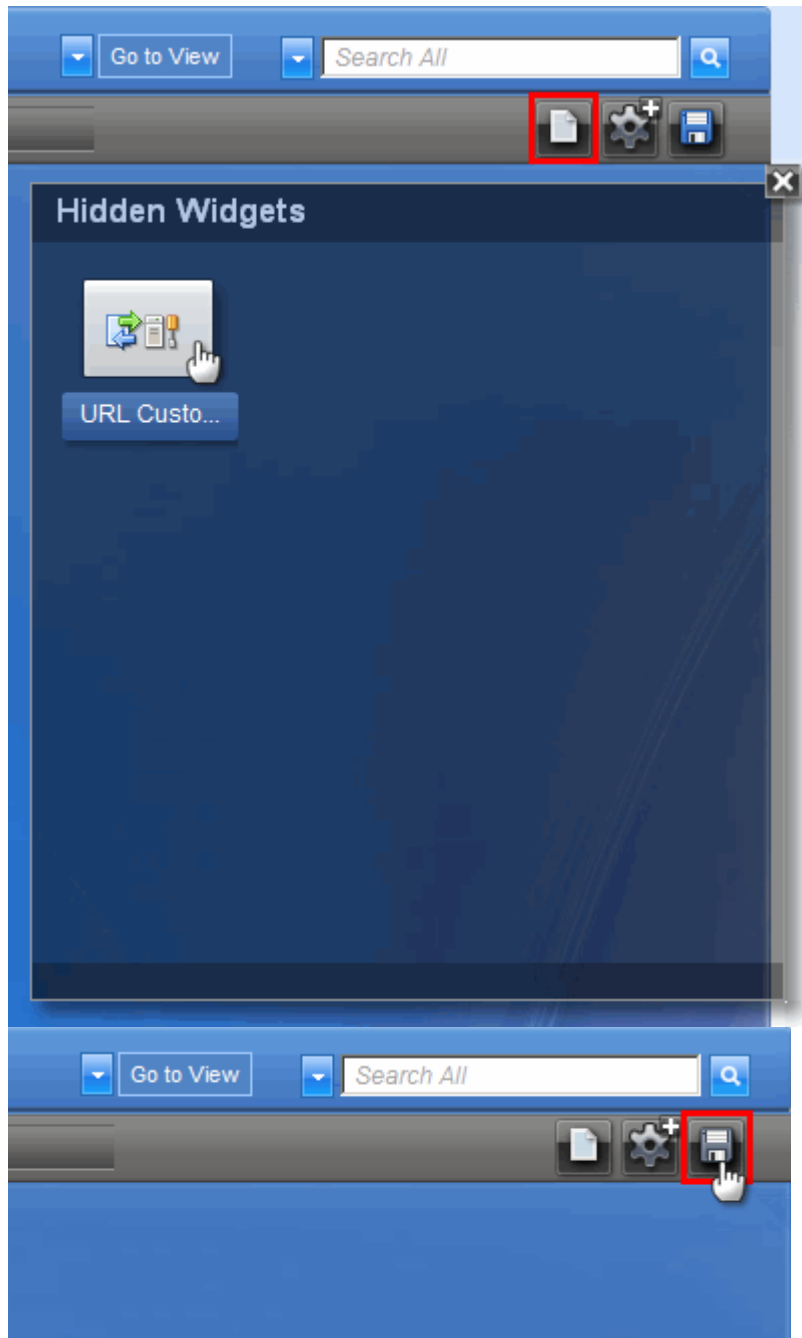


Figure 66. Saving your work

Figure 65. Viewing hidden widgets

28. Save your work by clicking the **Save** icon.

29. The mashup is now completed. It is time to try it out. Click **Go to View** to switch to the page viewing mode (clicking **Go to Edit** returns you to the editor).

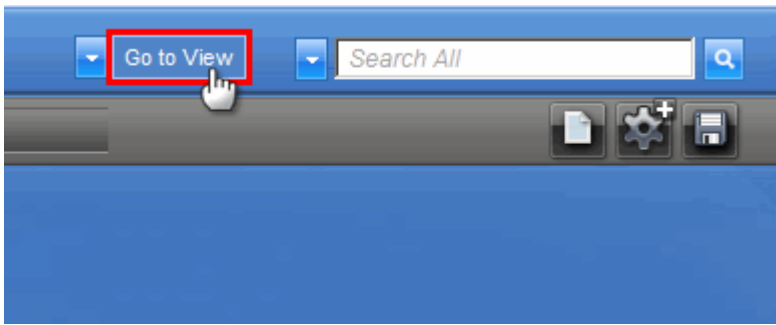


Figure 67. Switching to viewing mode

30. In the viewing mode, enter one of the names that we have in the sample database (such as “Simpson”) and click **Submit**. The record will be displayed in the Phonebook feed table.

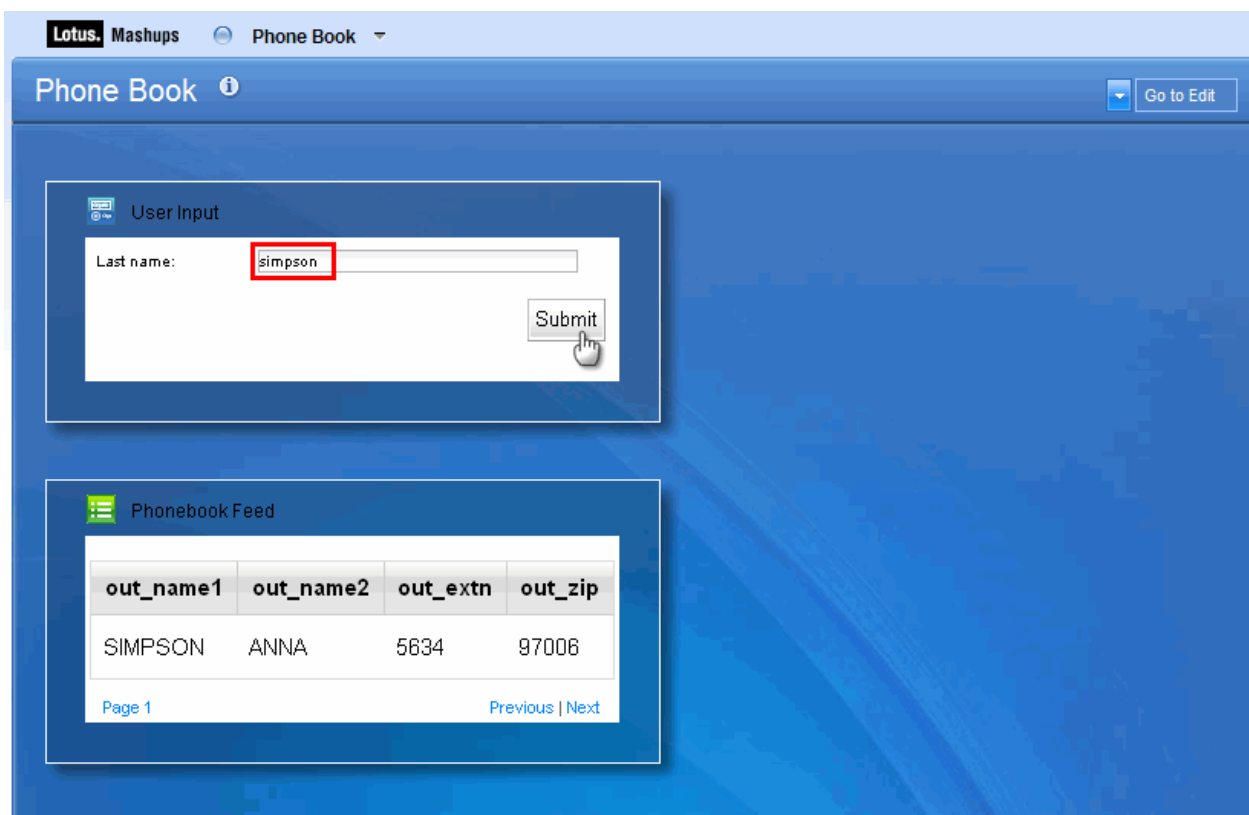


Figure 68. Testing the application

You have now created a simple web mashup using **IMS Phonebook** feed and other widgets.

The following optional steps let you experiment another widget, OpenStreetMap. You will add the **OpenStreetMap** widget to the page and send **ZIP code** to it from the **Phonebook** feed as the **Location**. When you left-click a person's zip code info in the **Phonebook** feed, the map widget will display the area around that zip code.

31. Go to Edit mode by clicking the **Go to Edit** button.
32. Expand the **Favorites** tab, mouse over the **OpenStreetMap** widget, and drag and drop the widget into the editor area.
You will now wire this widget with the Phonebook feed so the the zip code from the Phonebook Feed will become the input to the map.
33. Mouse over the OpenStreetMap widget and click the wiring icon with a green arrow and a blue arrow.
34. As shown in the following figure:
 - a. Click the **Receive** tab.
 - b. Under the Select content to receive section, click **Accept Location *using any Data***.
 - c. Under the Select a widget to send content section, click **Phonebook Feed**.
 - d. Under the Select content to send section, click the checkbox for **out_zip as Text**.
 - e. Click **Done**.

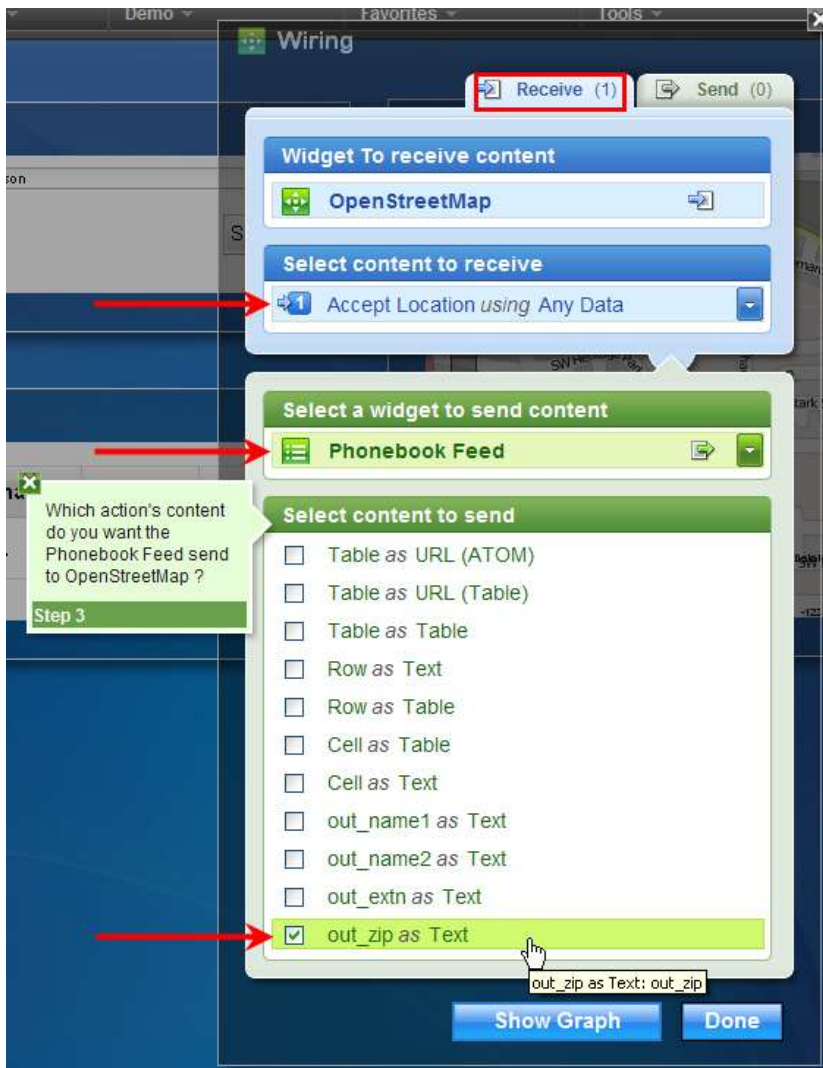


Figure 69. Wiring the OpenStreetMap with the Phonebook Feed

35. Click the zip code data (for example, 97006, for Anna Simpson) in the phonebook feed. The map widget is updated and shows the area around that zip code.

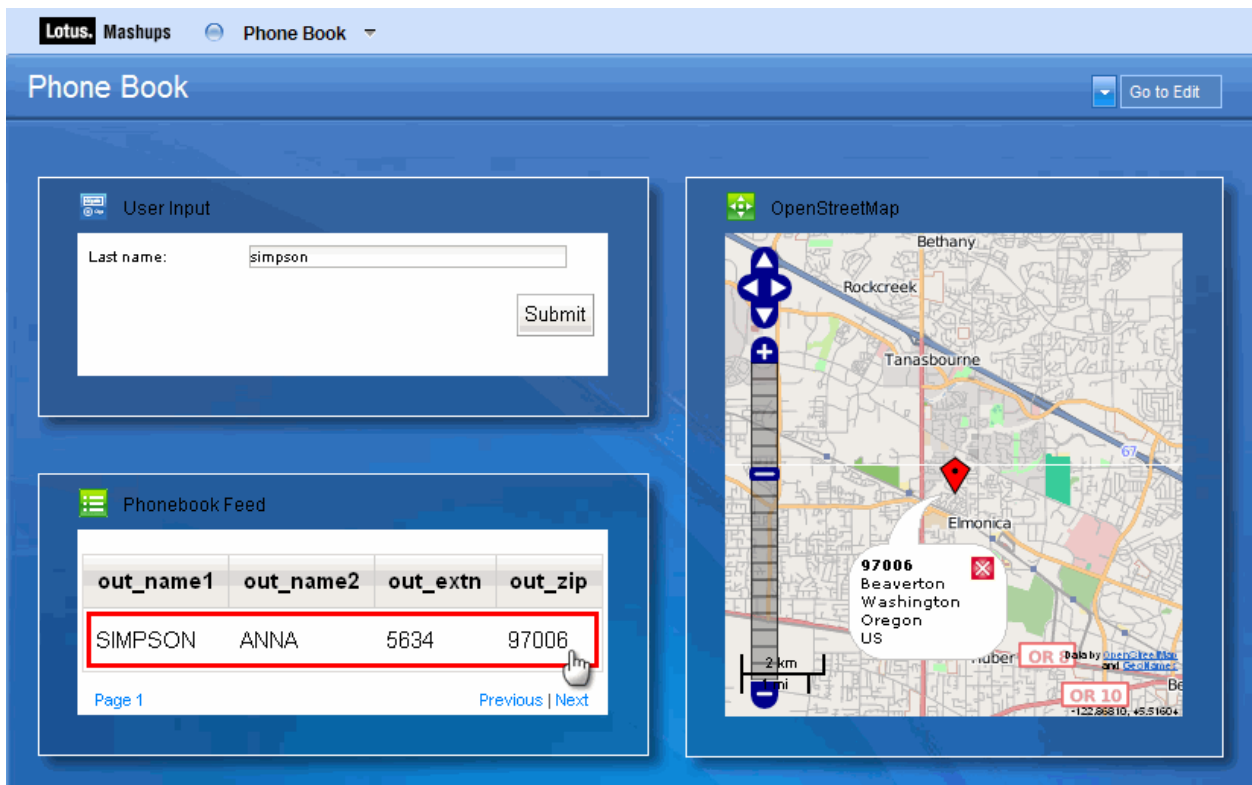


Figure 70. Adding OpenStreetMap widget

Important!

OpenStreetMap is a third party service and may not always be available

The next optional task shows how to improve the visual appearance of the **Phone Book** mashup. For example, the Phone Book table will have real column headers (such as **ZIP** instead of **out_zip**).

Task 3 (Optional). Improve the mashup page

Objective

The previous task created a mashup page that lets users specify a last name to look up in the phonebook. The displayed phonebook entry, however, has column headers that are not very friendly to feed users. As shown in Figure 70 at the end of Task 2, the data field names out_name1, out_name2, out_extn, and out_zip are not appropriate for our intended users.

In this task, we will create a better looking, more user-friendly phonebook feed by first using the operators in InfoSphere MashupHub to transform the data before we add the transformed feed to Lotus Mashups. We want to change the feed output to just three fields—Name (firstname + lastname), Extension, and Zip. The final page will look as follows:

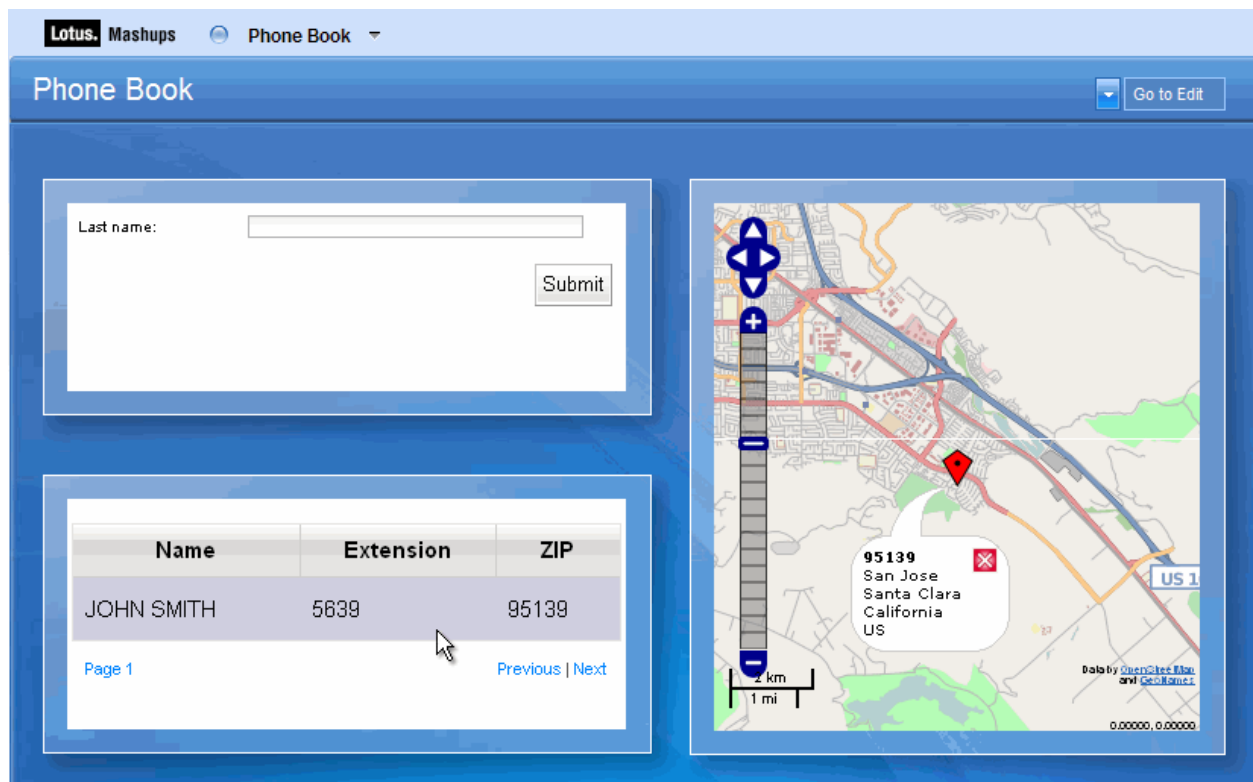


Figure 71. Final result

Step 1: Creating a Feed mashup

1. Go back to the **Home** tab of the **InfoSphere MashupHub** and select “**New Feed Mashup**”.

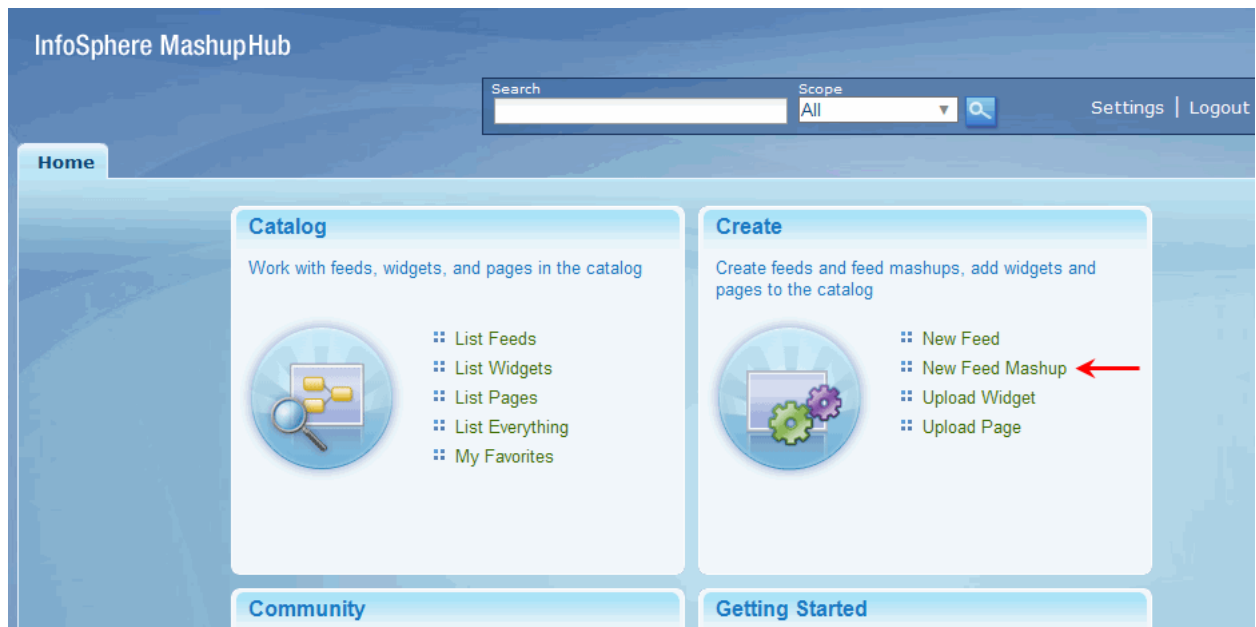


Figure 72. Selecting “New Feed Mashup”

A feed mashup is a feed that is created by taking one or more source feeds and applying operators and functions to filter and restructure the source data. MashupHub provides a feed mashup builder for creating feed mashups.

A **Feed Mashup editor** will open in a new tab, with a randomly generated feed mashup number. Note that the feed mashup number you see is likely going to be different from the one shown in the screen capture.

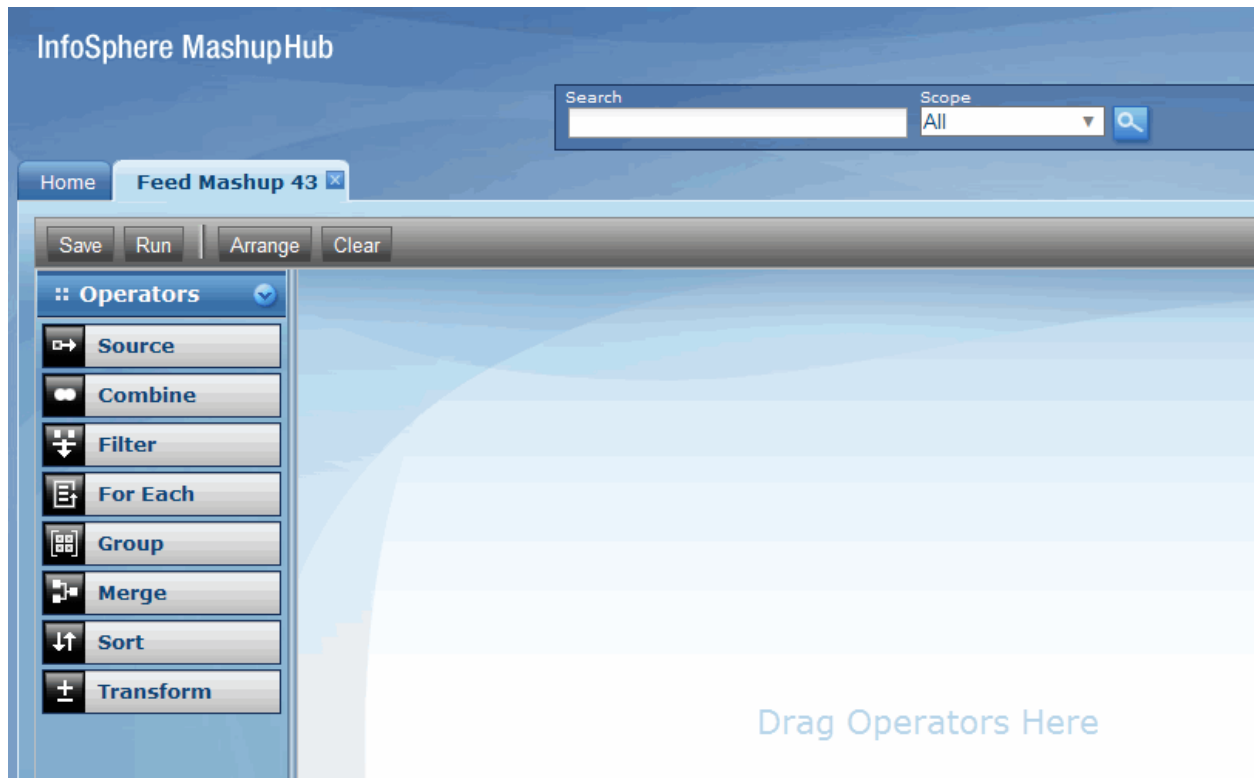


Figure 73. Feed Mashup Editor

2. Drag and drop the **Source** operator to the area that says “**Drag Operators Here**”. The “**Details for the Source operator**” window will automatically pop up.

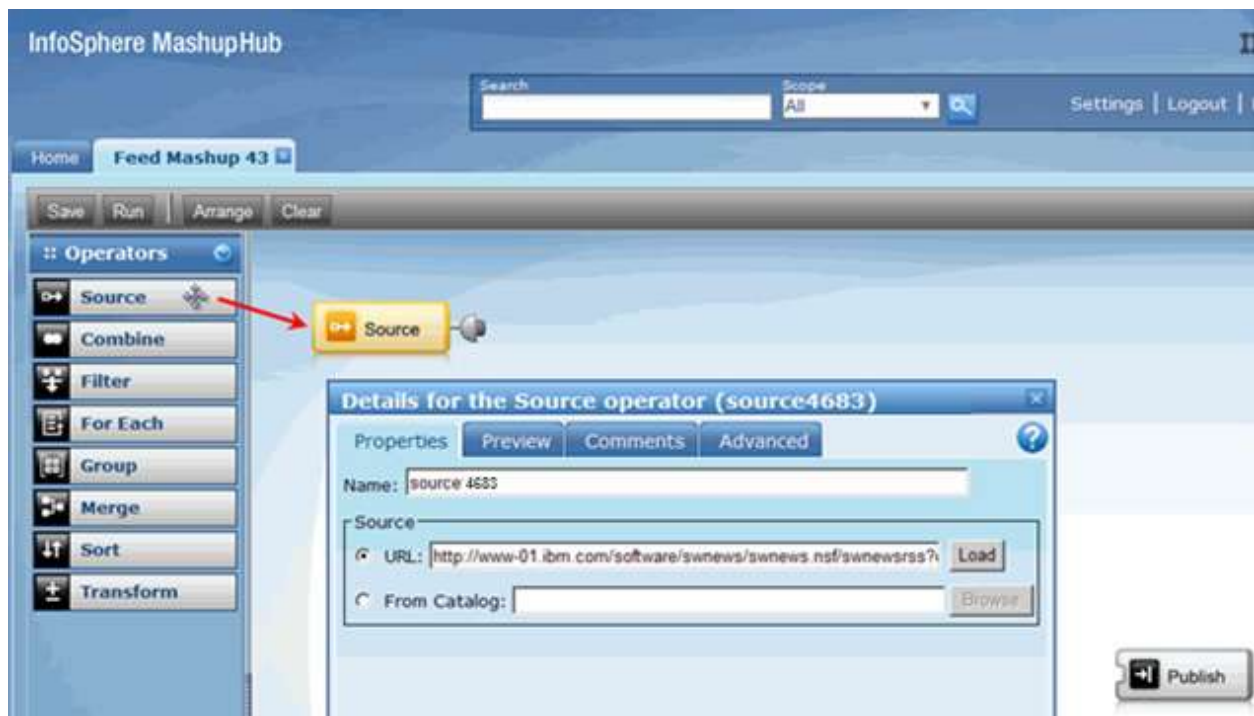


Figure 74. Adding “Source” operator

3. You can specify the URL of a feed or select it from the catalog. Let's select the feed from the catalog this time. Click "**From Catalog**" radio button and then click "**Browse**".

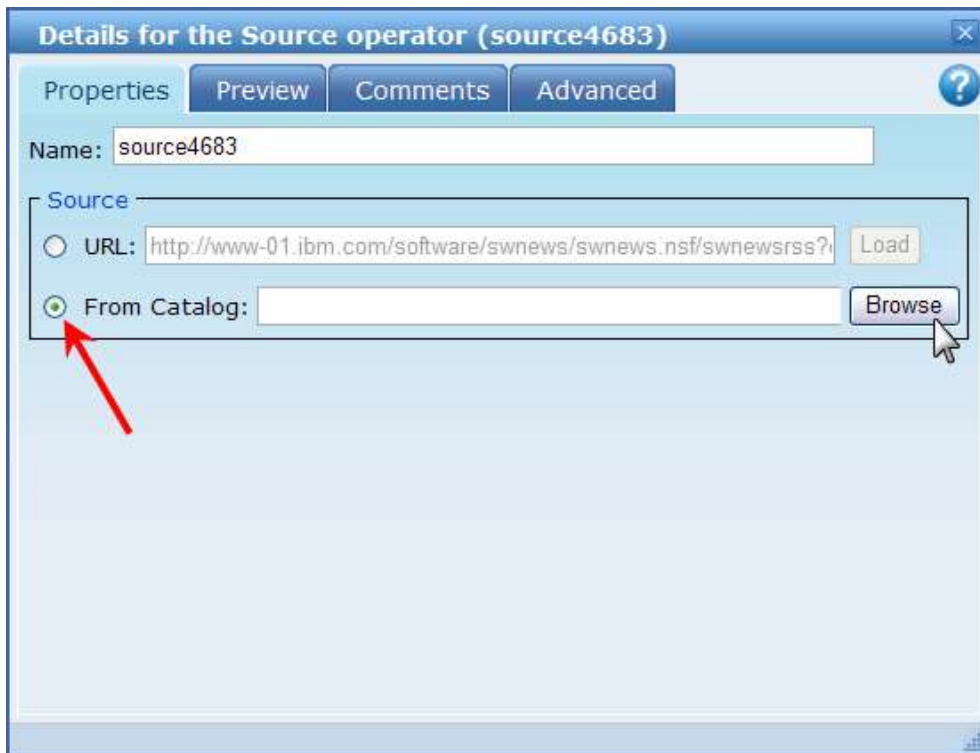


Figure 75. Selecting Feed Mashup source

4. Find **Sample Feed App** in the list (use arrow buttons to navigate the list). Click the **Sample Feed App** entry row and then click **OK**.

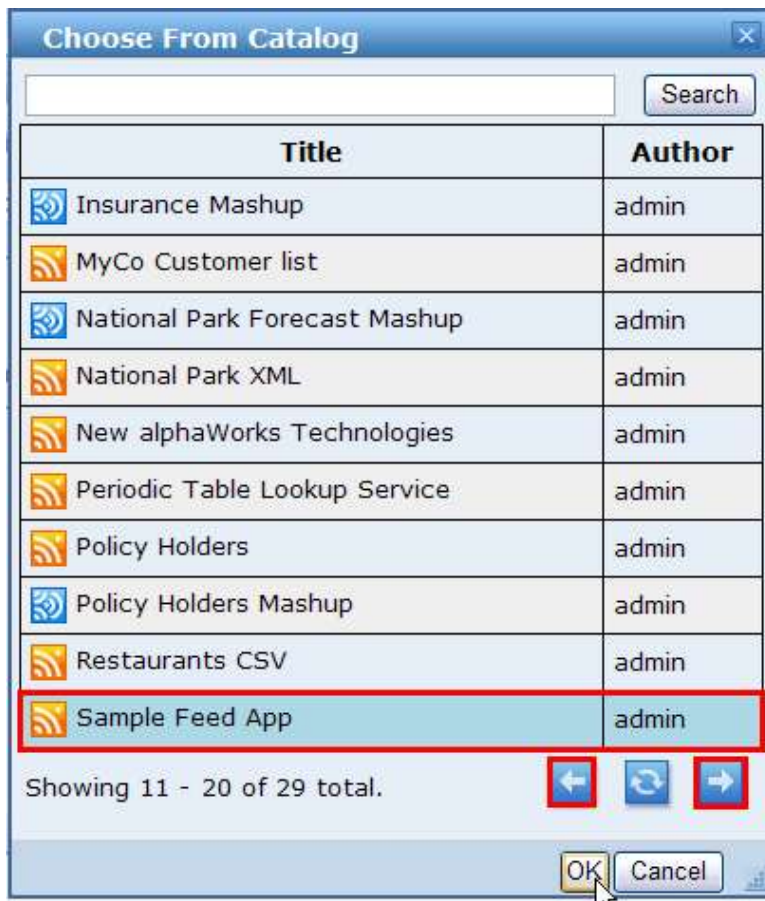


Figure 76. Selecting Sample Feed App

5. Click **Show URL Parameters**.

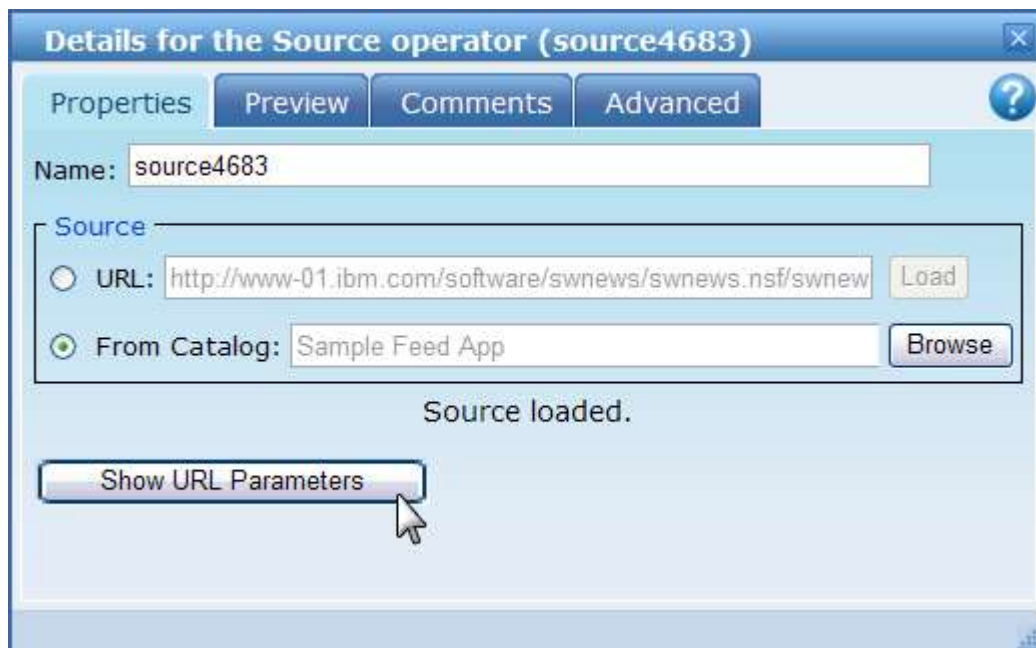


Figure 77. Selecting “Show URL Parameters”

A list of all the available URL parameters that are associated with the feed source displays.

6. Scroll the list of parameters all the way down.

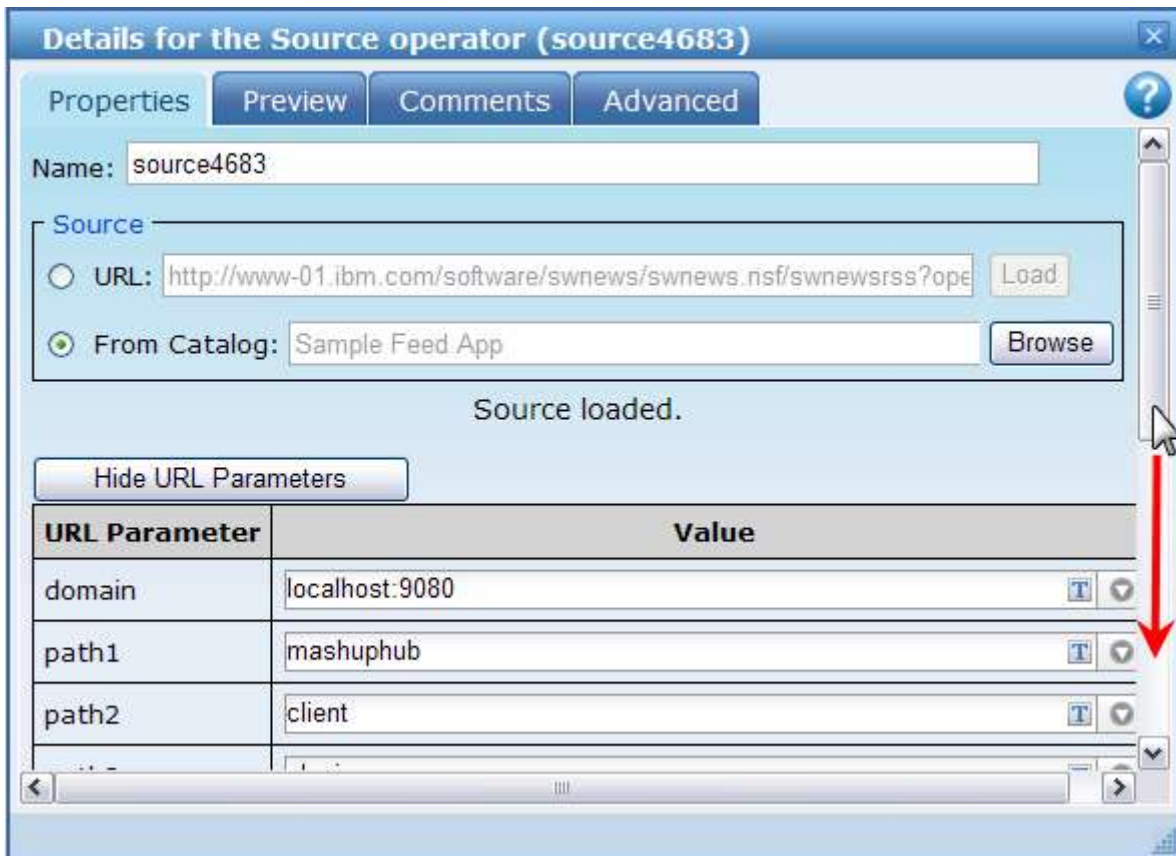


Figure 78. List of URL parameters

7. Click the grey drop down arrow next to “SMITH” and select the **Use a variable to return the value** option.

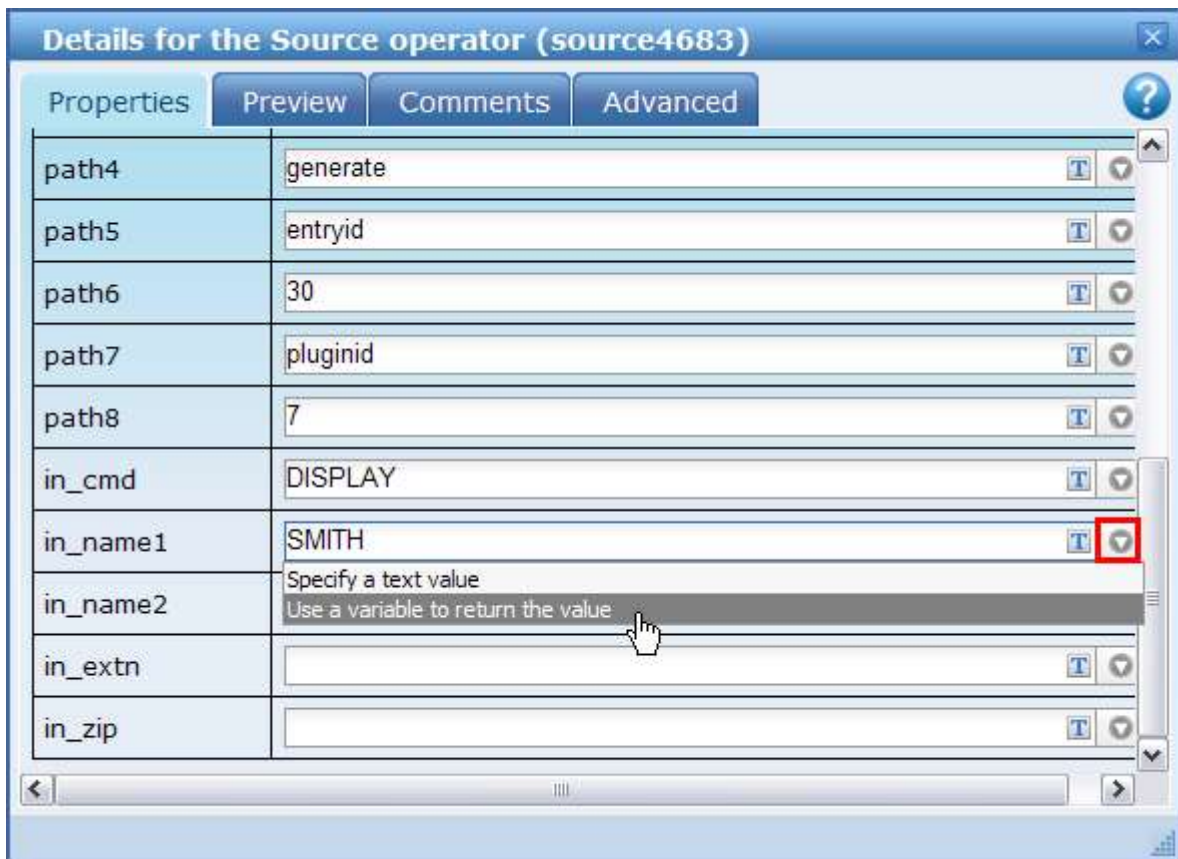


Figure 79. Defining return value as a variable

We chose “Use a variable to return the value” to turn this parameter into a variable.

8. Enter the word “lastname” (no spaces) in the **Name** field and “SMITH” in the **Default Value** field. Click **OK**.

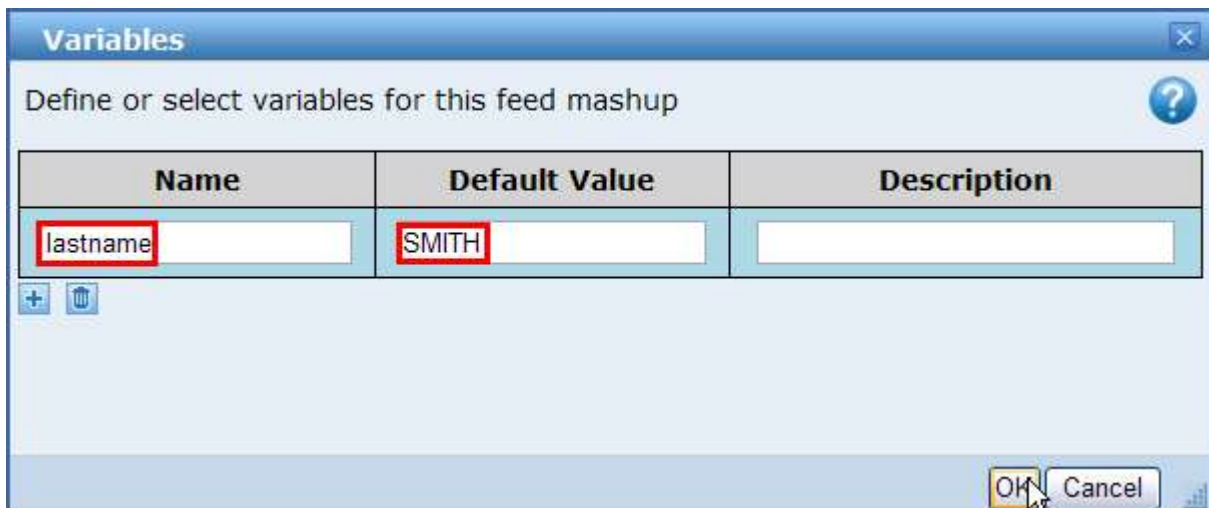



Figure 80. Defining a variable

9. Close the “**Details...**” screen by clicking  button.

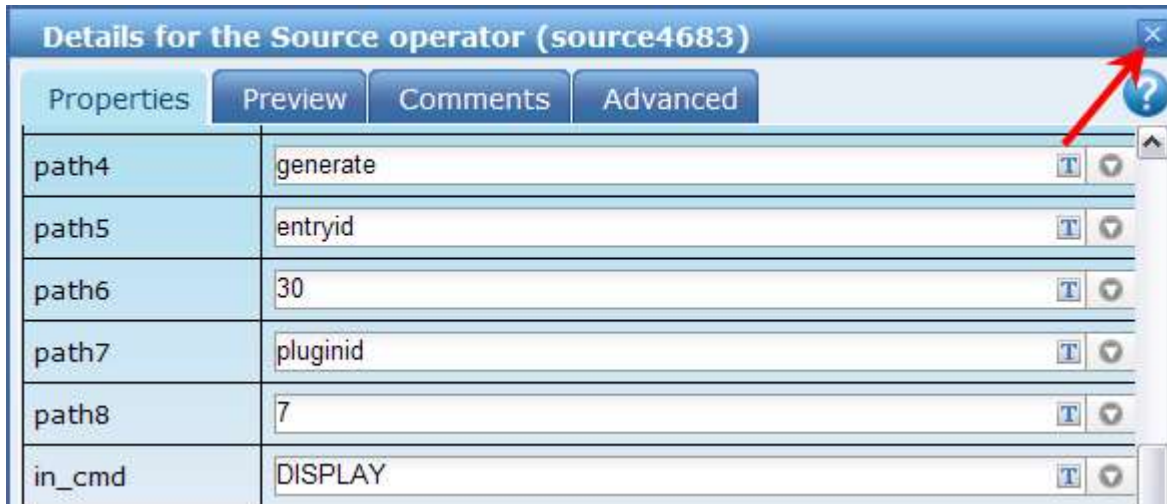


Figure 81. Closing “Details...” screen

10. Drag and drop the **Transform** operator into the area between the **Source** and the **Publish** operators.

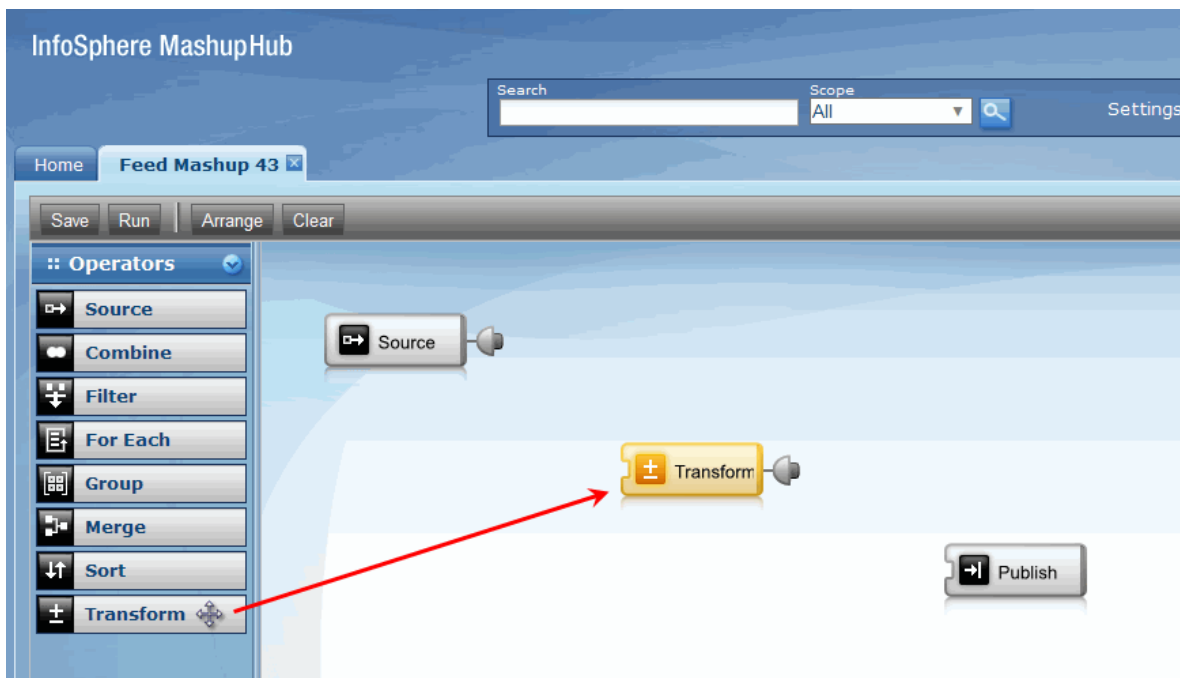


Figure 82. Adding Transform operator

The Transform operator lets you restructure an incoming feed. You can create new elements, remove elements, add attributes to elements, apply a mathematical or text function to an element, and manipulate the values of the elements.

We will use the Transform operator to transform the data coming from the feed (out_name1, out_name2, out_ext, and out_zip) to something that’s more user-friendly (Name, Extension, and ZIP) in the actual output on the mashup page.

11. Connect operators by dragging and dropping the plug from the **Source** to the **Transform** and from the **Transform** to the **Publish** operator.

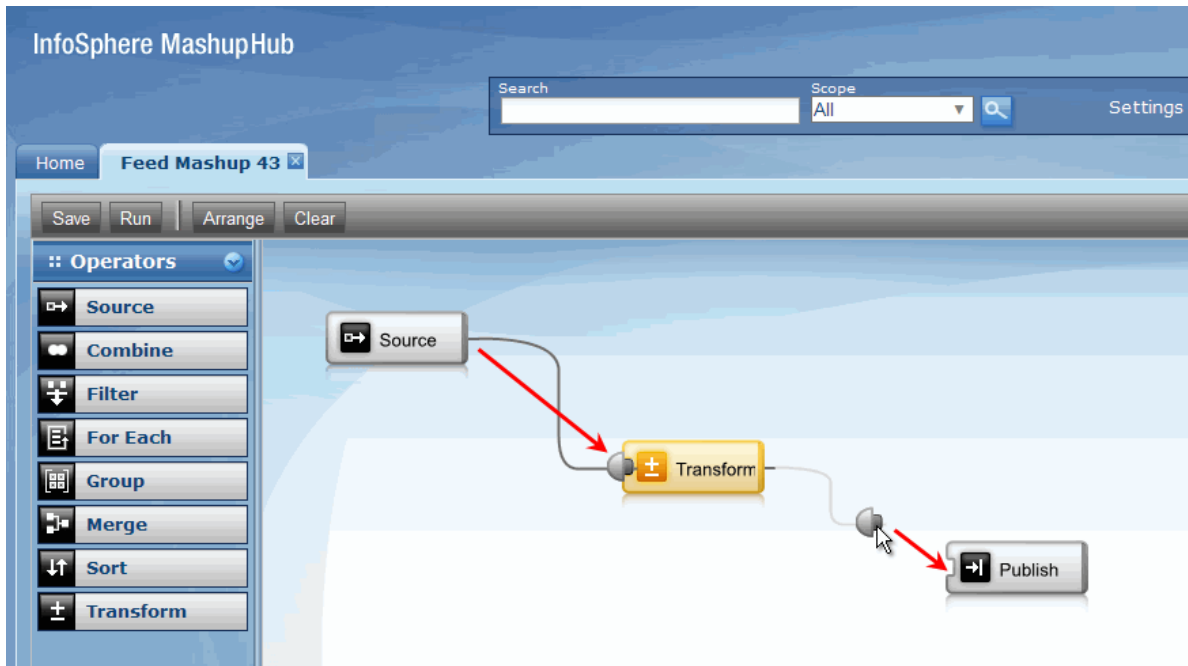


Figure 83. Connecting operators

12. Click the **Transform** operator to edit its details.

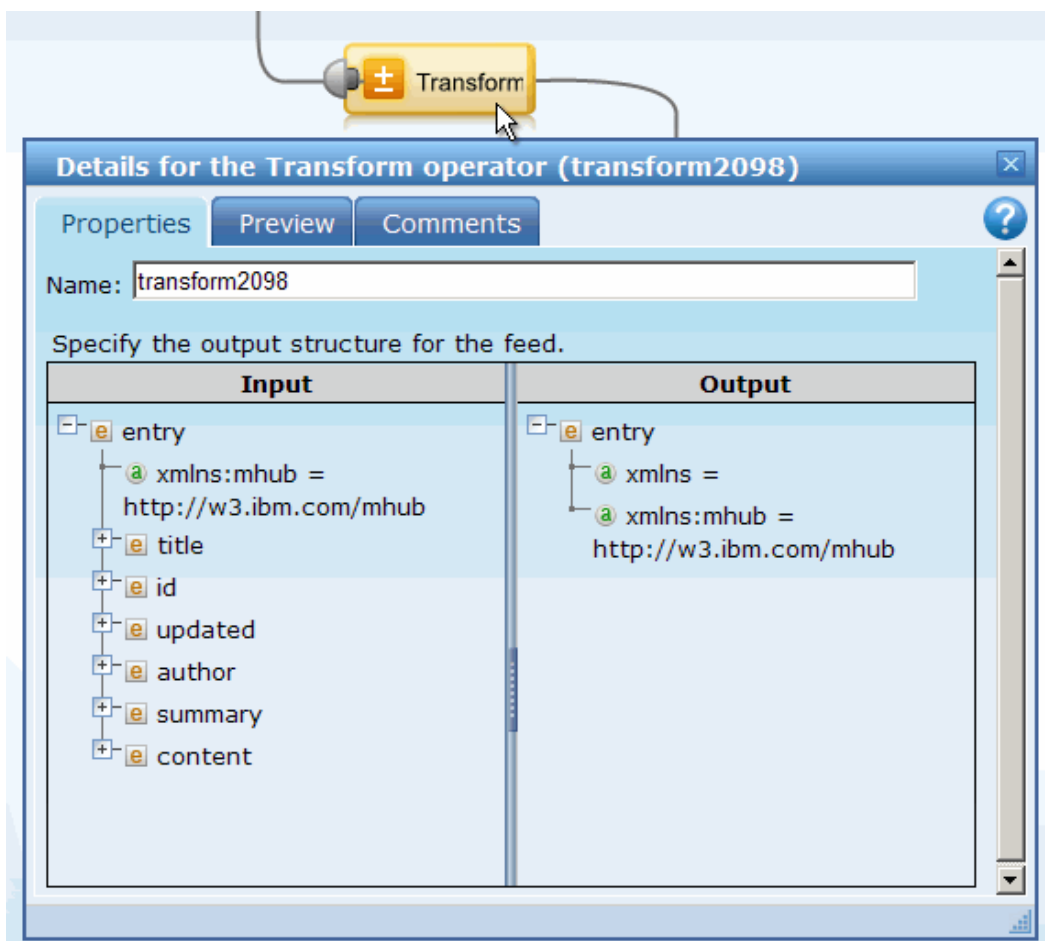


Figure 84. Editing the details of the Transform operator

Generate an IMS feed using IMS Web 2.0 Solution for InfoSphere MashupHub

© Copyright IBM Corporation 2009. All rights reserved.

We want to first create the output structure that we want our users to see, so we will first create new elements in the Output tree and then specify how these elements get their values from the Input tree.

XML structure of a feed

Your feed output must have a “content” element to hold the feed content. Under this content element, you then define the sub-elements to hold the data for each field.

13. Create a new element in the Output tree. Right click on the **entry** element in the **Output** panel and select **New Element**.

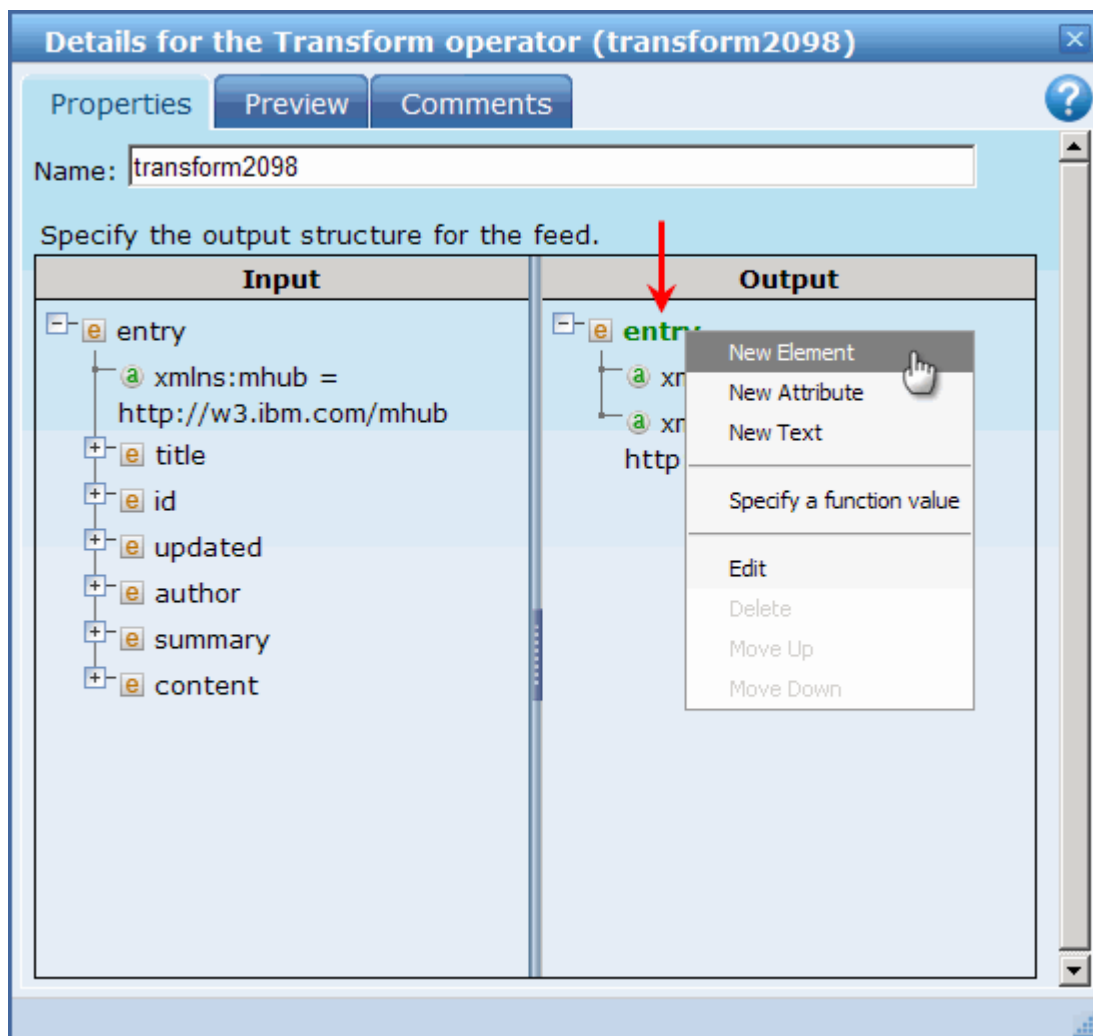


Figure 85. Selecting New Element

14. Enter “**content**” as the name of the new element.

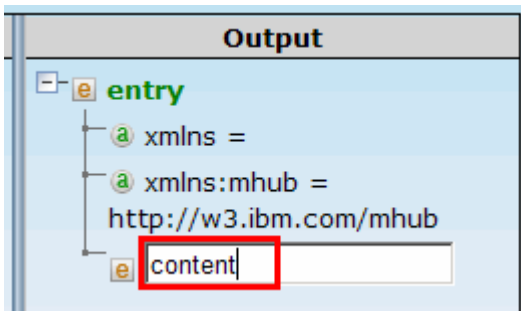


Figure 86. Creating ‘content’ element

15. Create a new element called “**Name**” under **content** element. Right click **content**, select **New Element** and call it “**Name**”.

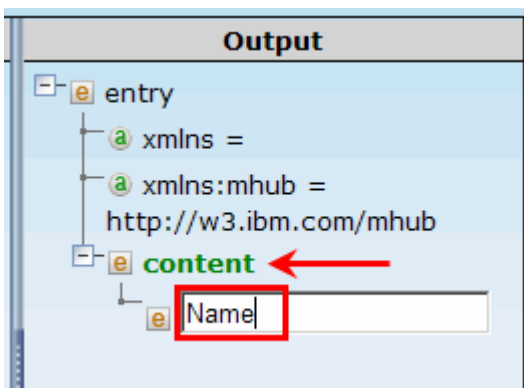


Figure 87. Creating ‘Name’ element

16. Using the same procedure, create two more elements – “**Extension**” and “**ZIP**”.

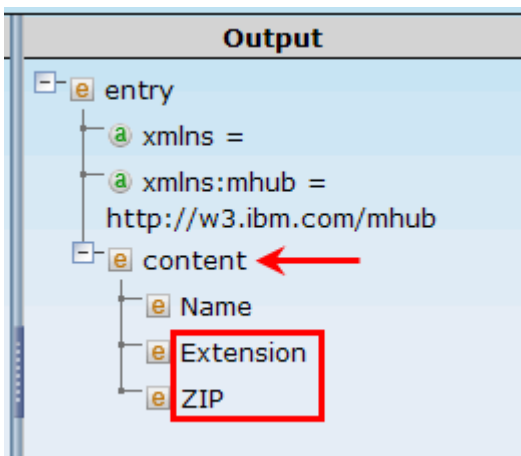


Figure 88. Creating ‘Extension’ and ‘ZIP’ elements

17. Right click the “Name” element and select **Specify a function value**.

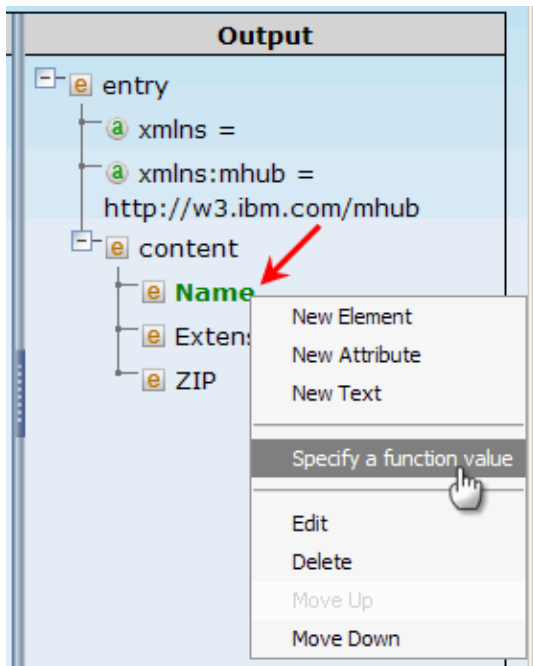


Figure 89. Selecting ‘Specify a function value’

18. In the function selection dialog, choose **Concat** as the function to perform.

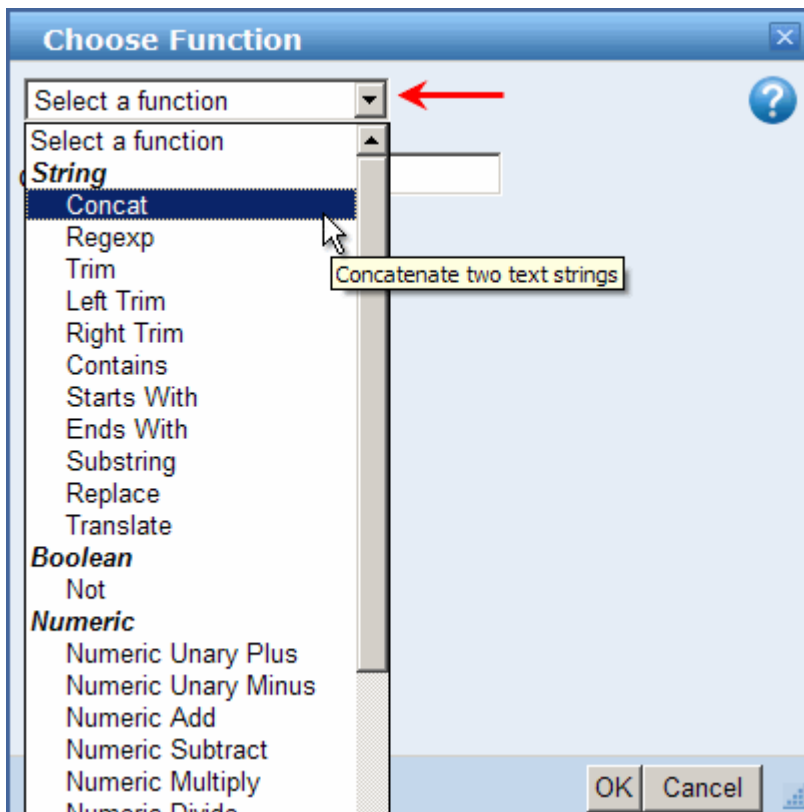


Figure 90. Selecting Concat function

19. Expand the drop down menu for the **First String** and select **Specify a value from the Input tree**.

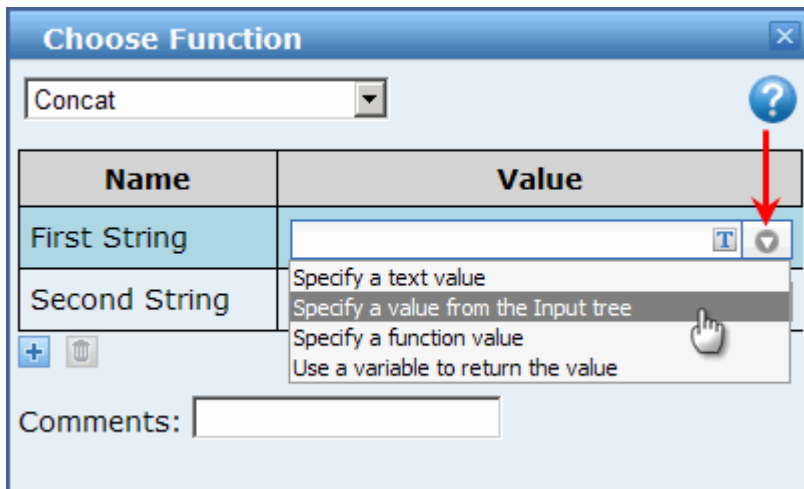


Figure 91. Selecting “Specify a value from the Input tree”

20. In the Input tree display window, expand content, RepeatingElement and OUTPUTMSG elements, then select out_name2 and click OK.



Figure 92. Selecting out_name1

21. For the **Second String**,
 - a. Expand the drop down menu for Second String and select **Specify a text value**.

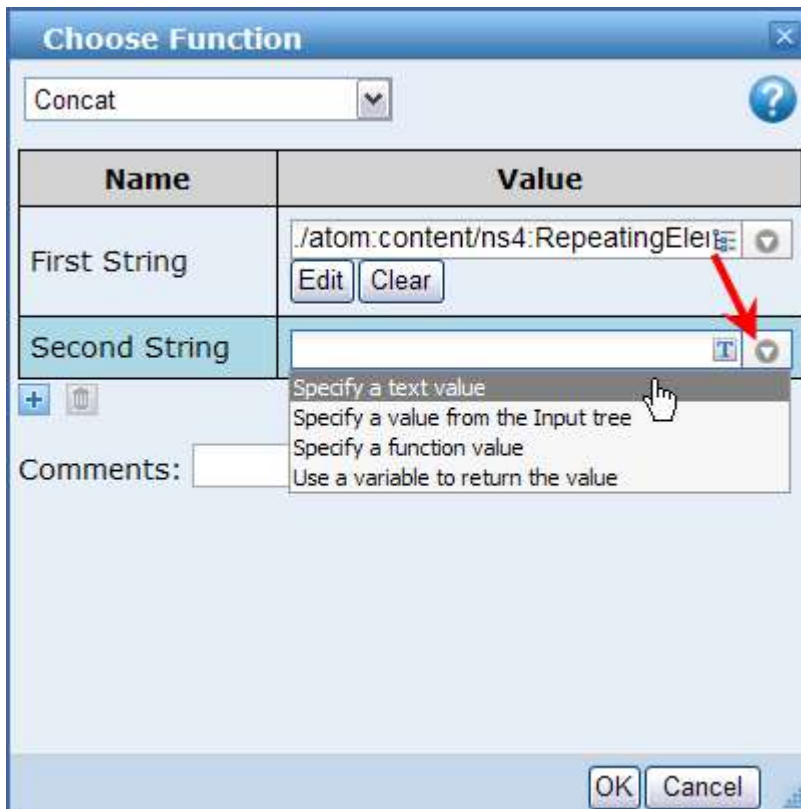


Figure 93. Set this second string to be of text value

- b. Enter quotation marks with a single space in between (“ ”), as shown in the following figure. This will be the delimiter between the first and the last name. Then click **+** to add one more string to be concatenated.

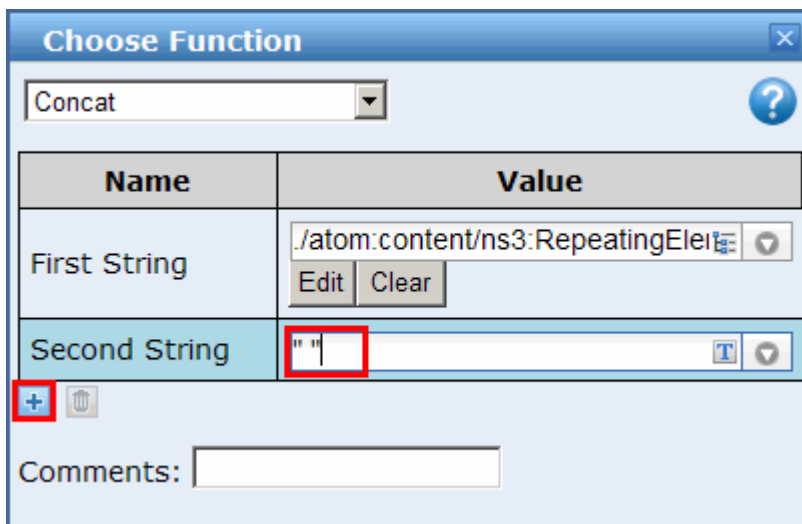


Figure 94. Specifying space character as a delimiter

22. For the new **Custom** string, select “Specify a value from the Input tree” again.

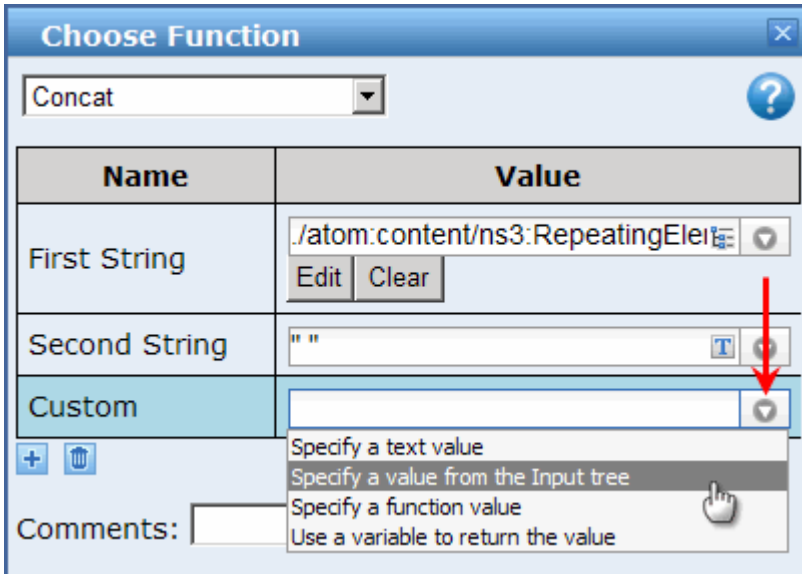


Figure 95. Selecting “Specify a value from the Input tree”

23. In the Input tree display window, expand **content**, **RepeatingElement** and **OUTPUTMSG** elements, then select **out_name1** and click **OK**.

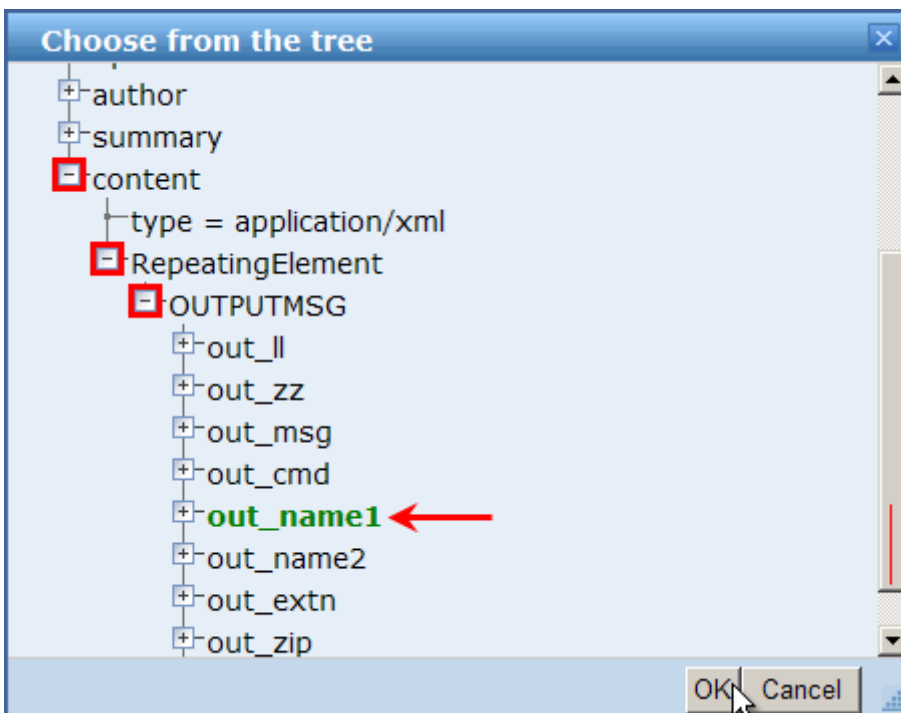


Figure 96. Selecting out_name2

24. Click **Ok** to close **Function** dialog.

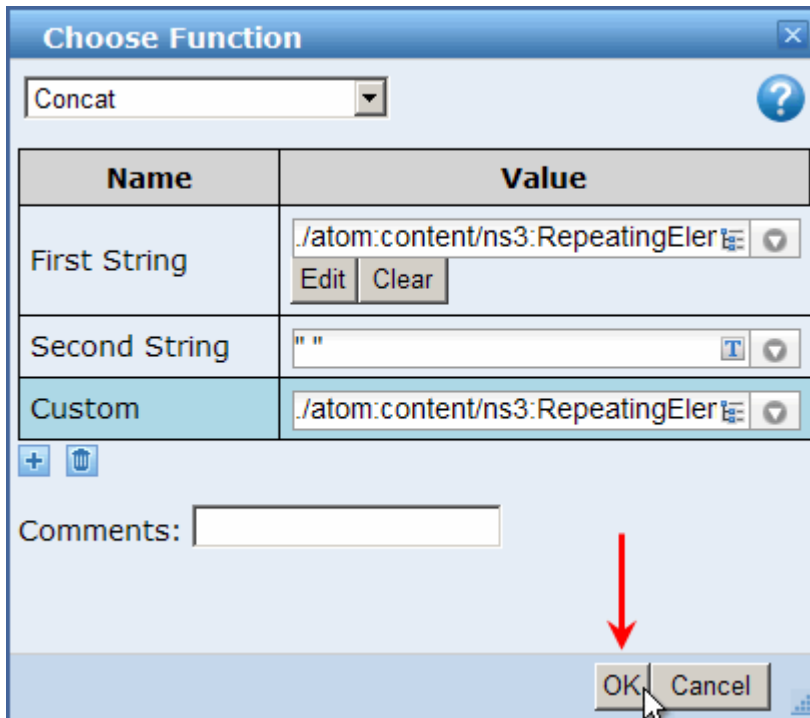


Figure 97. Closing function dialog

25. Expand the elements of the **Input** tree on the **Details** screen, as shown in the following figure.

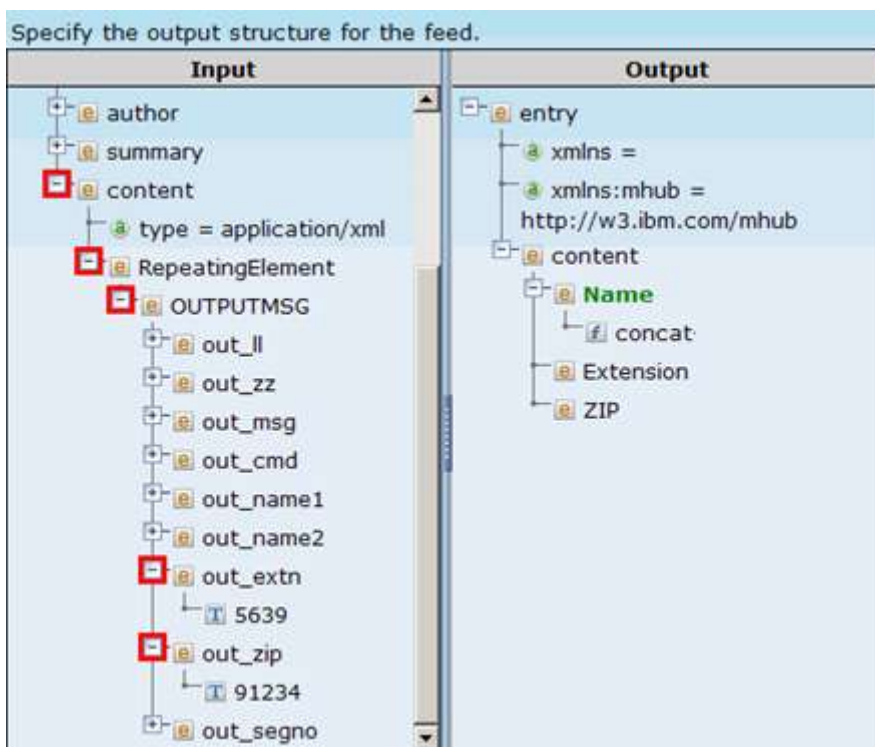


Figure 98. Expanding input tree elements

26. Drag and drop “5639” onto the “Extension” element in the **Output** tree, drag and drop “91234” onto the “ZIP” element.

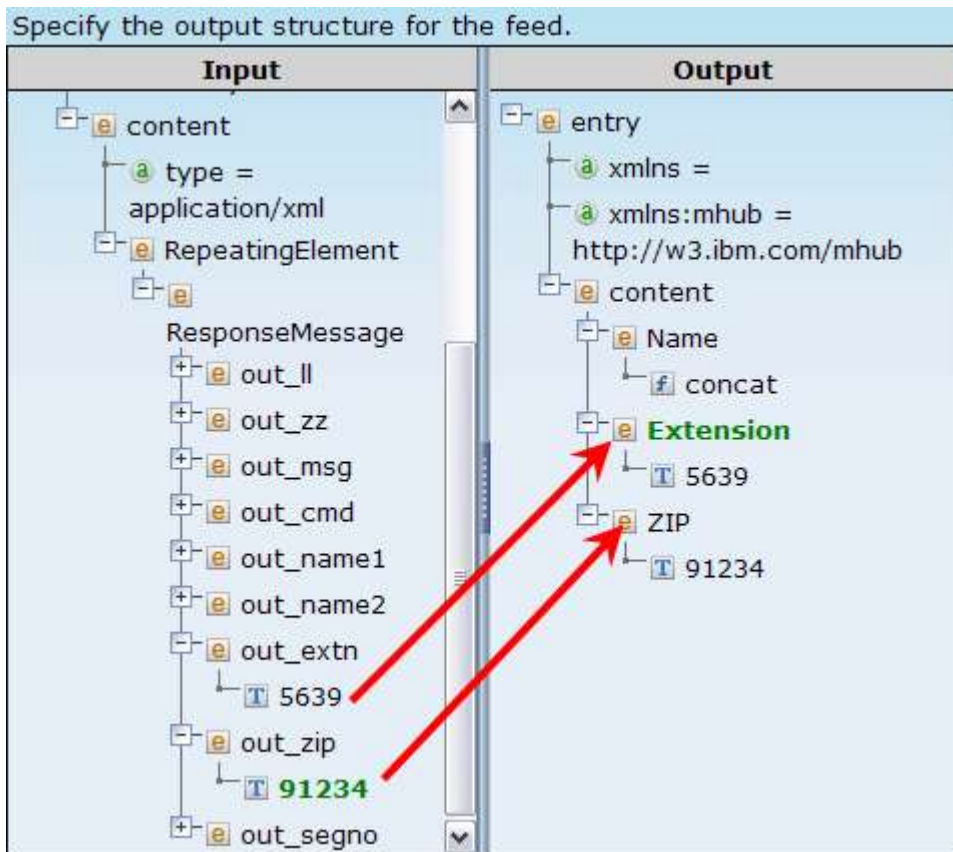


Figure 99. Dragging and dropping elements

27. Close the “Details for the Transform operator” window.

28. Click the **Publish** operator to open its “Details..” screen. Select **ATOM** as the **Feed Type**.

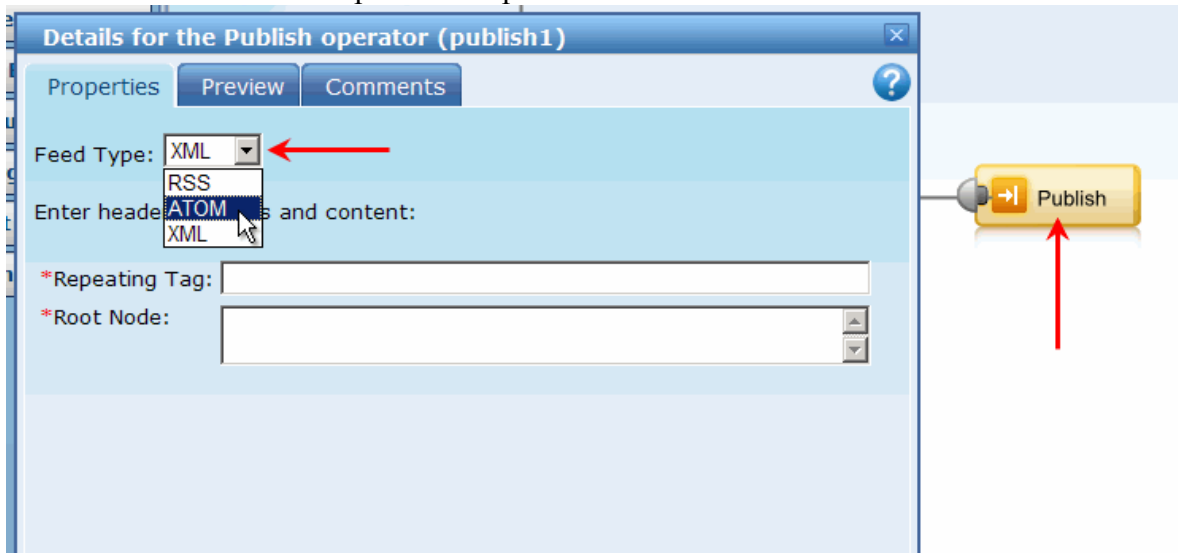


Figure 100. Selecting ATOM feed type

29. If you get a message saying the data is out of sync, click the **Refresh** button.

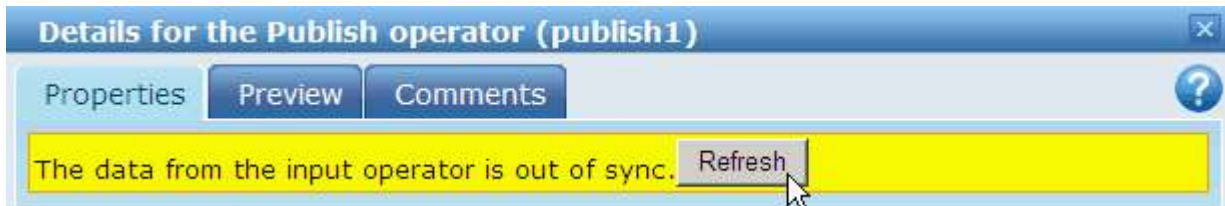


Figure 101. Refreshing the data

30. Provide required header elements, such as **Title** and **Author** the close the **Details** dialog.

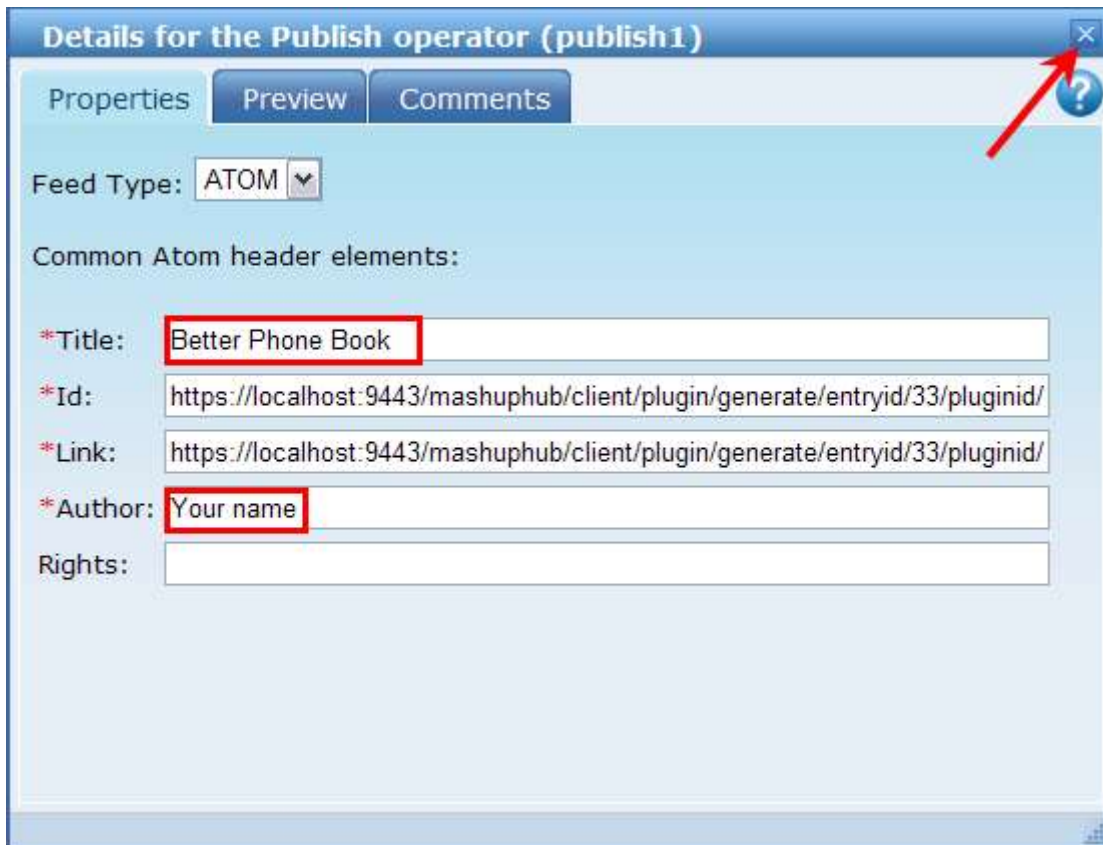


Figure 102. Entering required header elements

31. Click the **Save** button on the left side of the Feed Mashup editor.

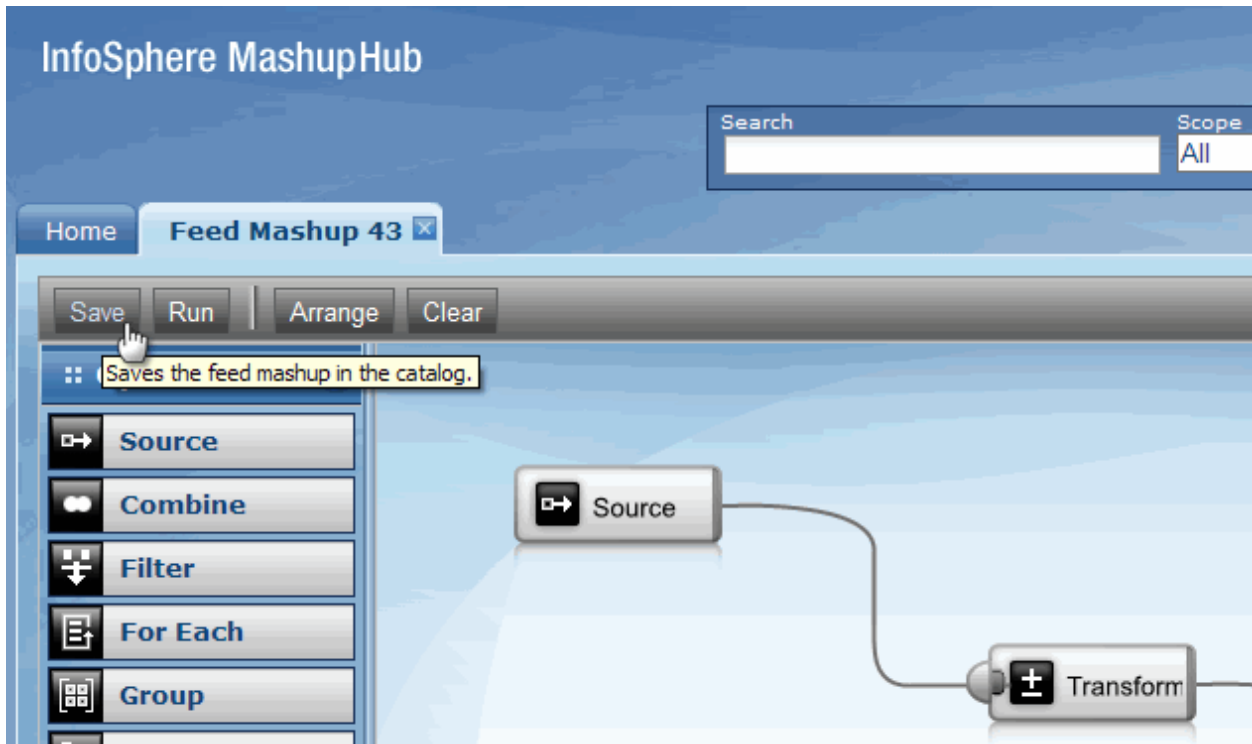


Figure 103. Saving feed mashup

32. Enter required information as shown in the following figure and click **Finish**.

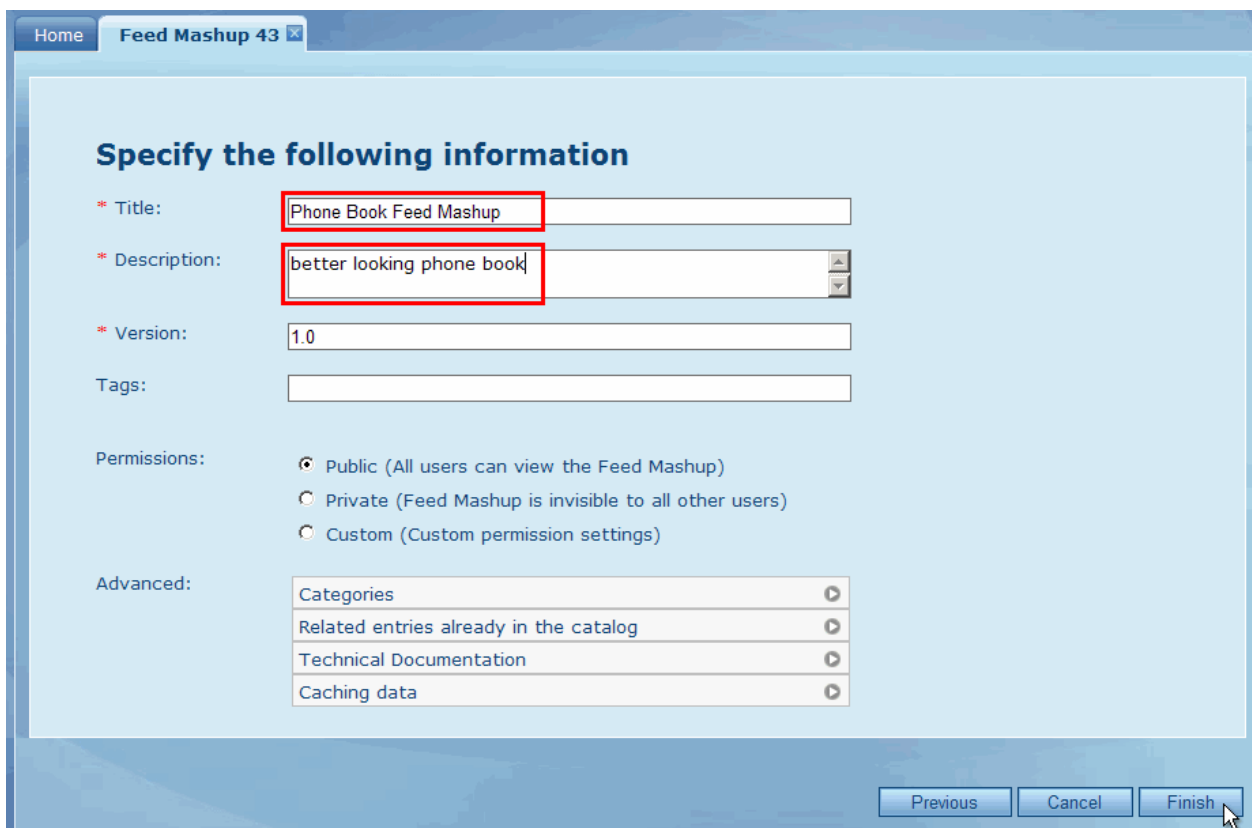


Figure 104. Entering required information

Now we need to change the **Phonebook Feed** widget in **Lotus Mashups** so that it displays the new better looking phonebook. Like noted earlier in the tutorial (Task 2, Step 1, sub-step 8), we do not need to add the new feed to **Lotus Mashups** and replace the old one with it. Instead, since we used the generic **Data Viewer** widget, we simply need to provide the URL of the new feed to the **Data Viewer**.

33. Go back to the **InfoSphere MashupHub** Home screen and select **List Feeds**.



Figure 105. InfoSphere MashupHub home screen

34. Find the “**Phone Book Feed Mashup**” in the list and select it. While on the **Details** screen, select and copy (Ctrl-C) the feed URL.

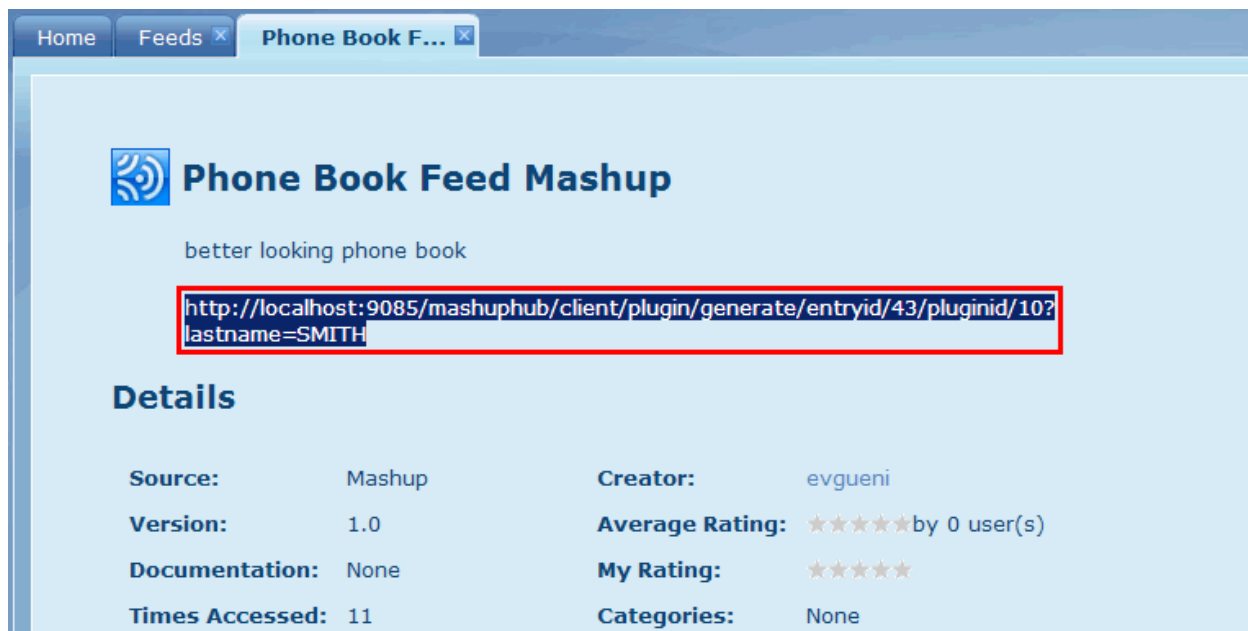


Figure 106. Copying the feed URL

Since widgets in our application are wired in such a way that the **Data Viewer** widget (**Phonebook feed**) receives data via the URL provided by the **URL Customizer**, we actually need to replace the **URL Customizer**'s input URL with the new one.

35. Go back to **Lotus Mashups**. If you are in View mode (you will see the **Go to Edit** button), click the **Go to Edit** button to go to Edit mode.



Figure 107. Going to Edit mode

36. Click the **Hidden Widgets** icon.

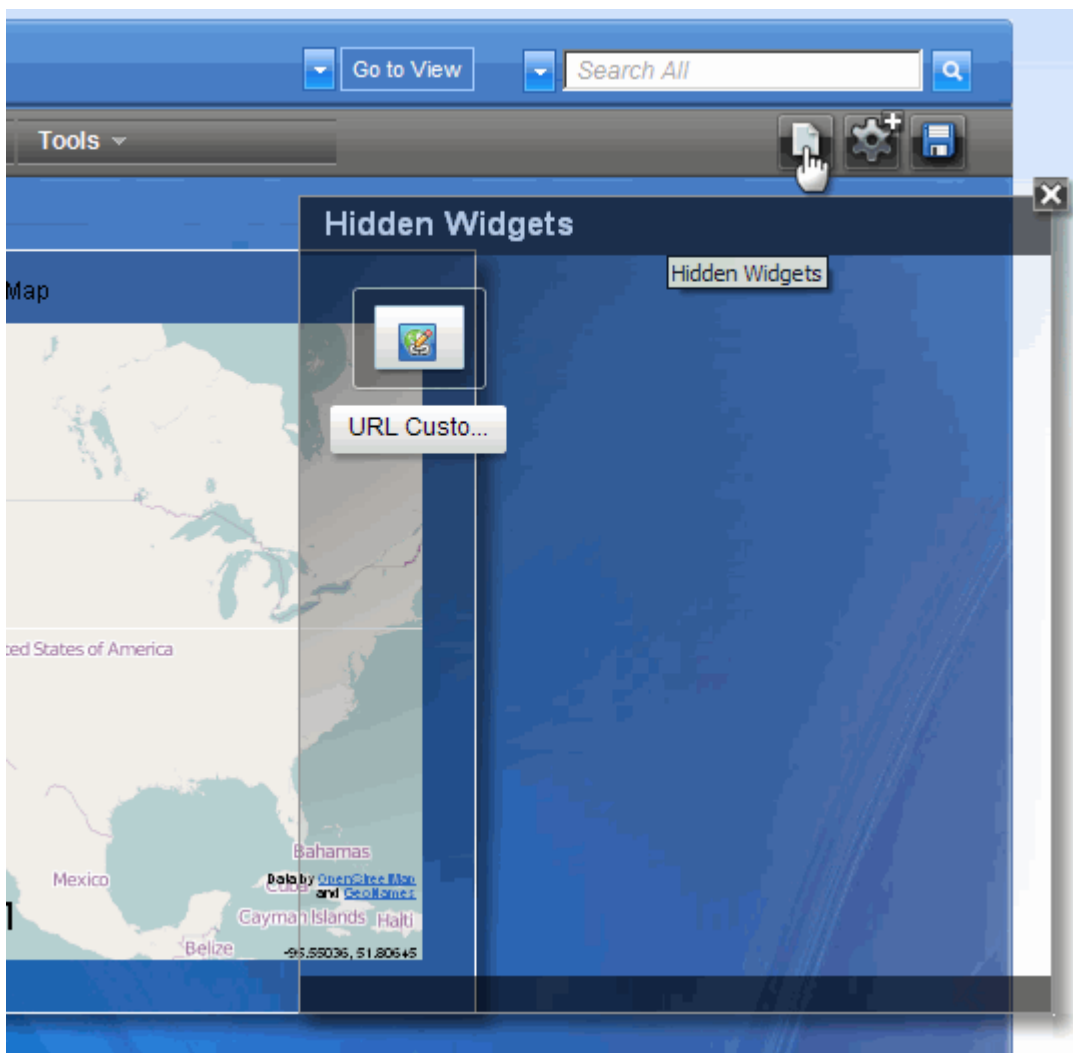


Figure 108. Revealing hidden widgets

37. Click the **options** menu of the **URL Customizer** and select “**Edit Settings**”.

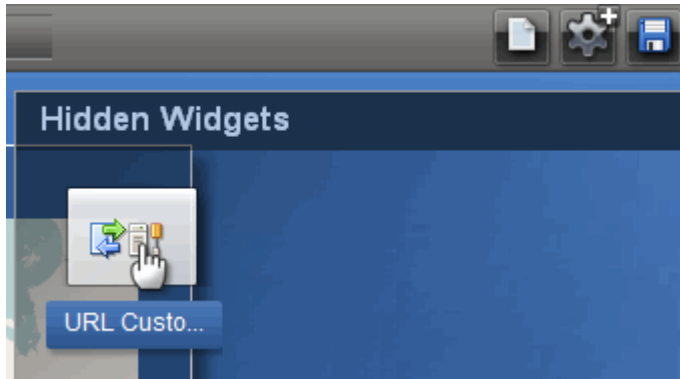


Figure 109. Selecting URL Customizer options menu

38. Delete the existing URL and paste (CTRL-V) the URL of the new **Phone Book Feed Mashup**. Click **Load**. Since this new feed only has one parameter (**lastname**), it will be selected automatically, so just click **Save** to close the dialog.

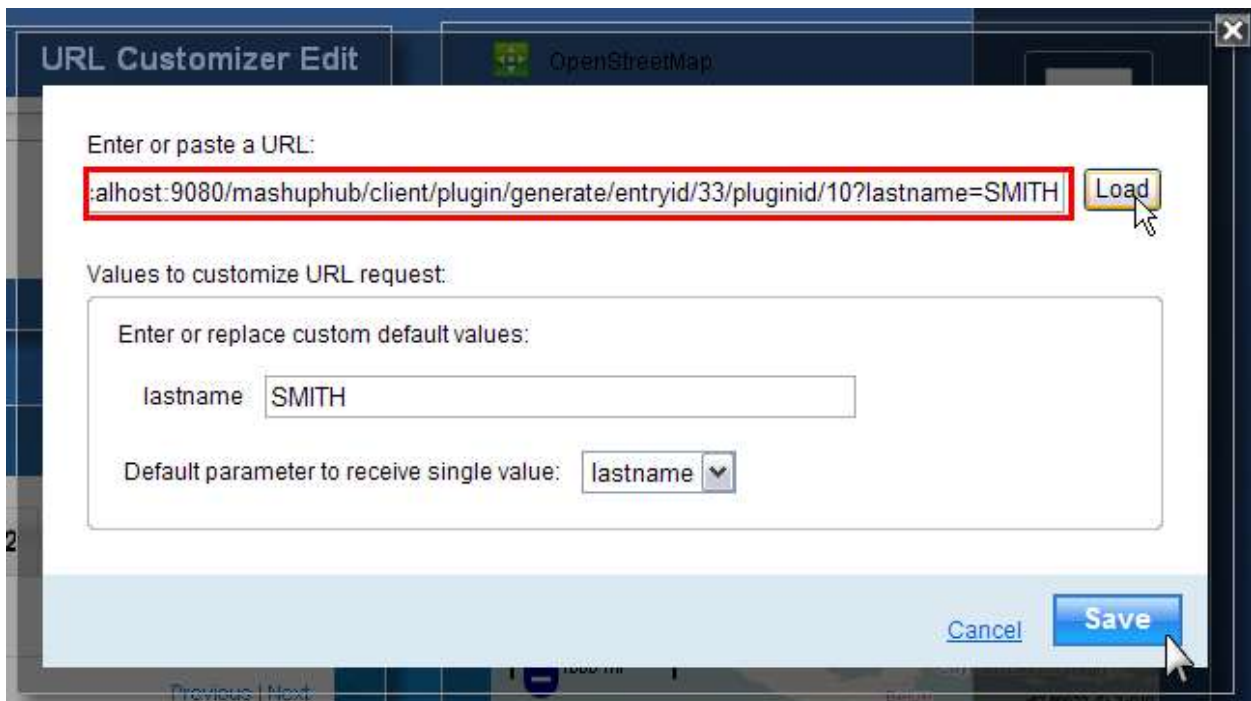


Figure 110. Replacing feed URL in URL Customizer

The **Phonebook Feed** widget will get automatically refreshed with new contents. Notice the new and improved column names and table format.

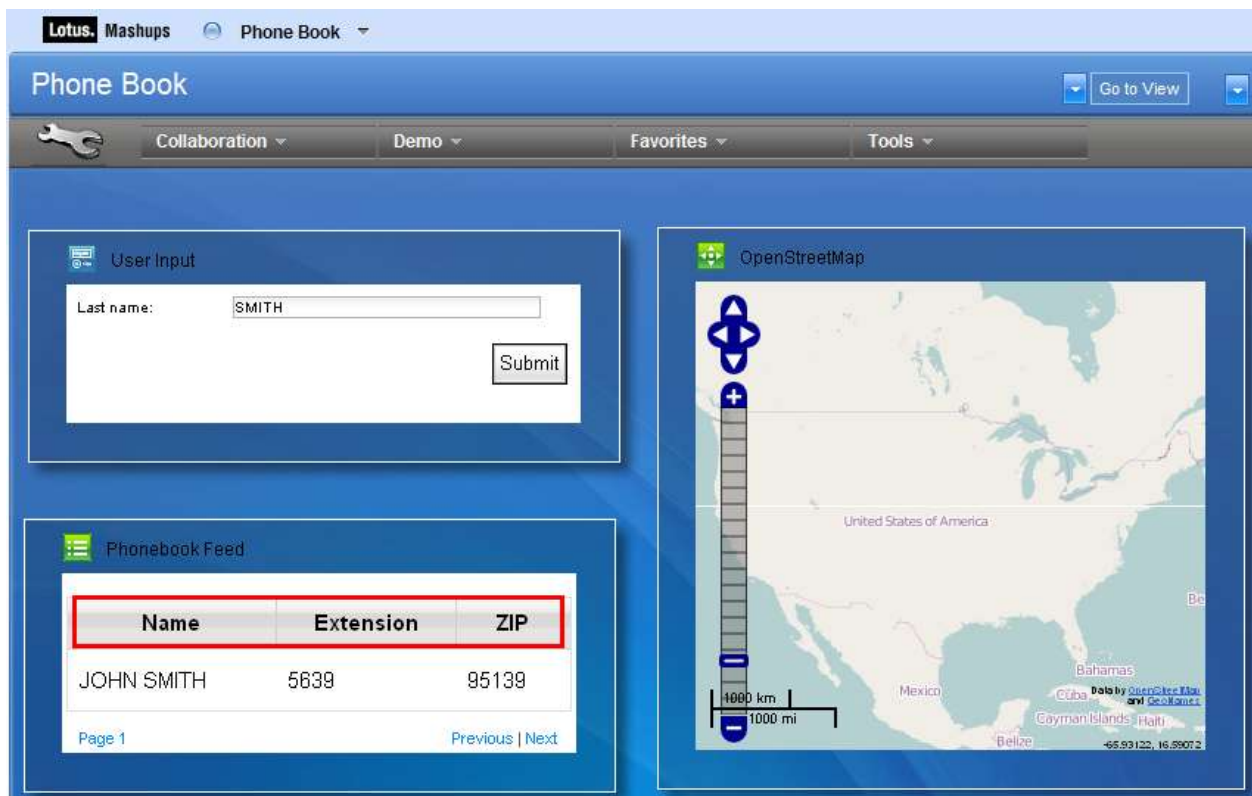


Figure 111. New Phonebook Feed widget

39. In the top right corner of the editor pane you can find the **page layout options icon** (it will only appear when your mouse cursor is within the editor area). Click the icon to open **page layout options** dialog.

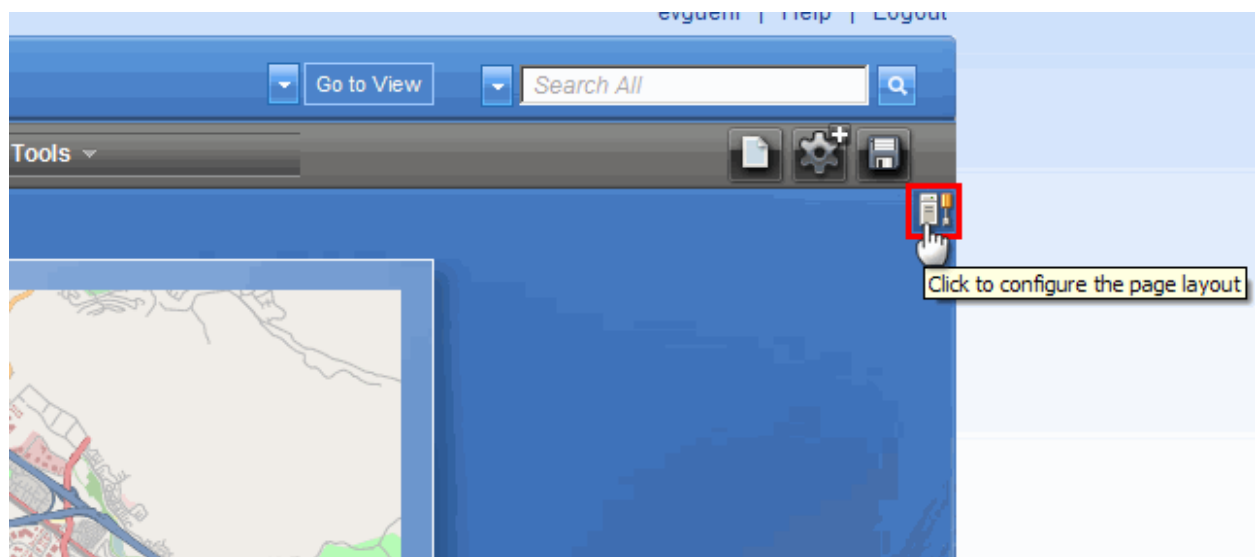


Figure 112. Page layout options icon

40. Select a different default skin that will apply to all widgets and click **Save**. Observe how the visual appearance of your application changes. Try different skins to see which one suits your taste better.

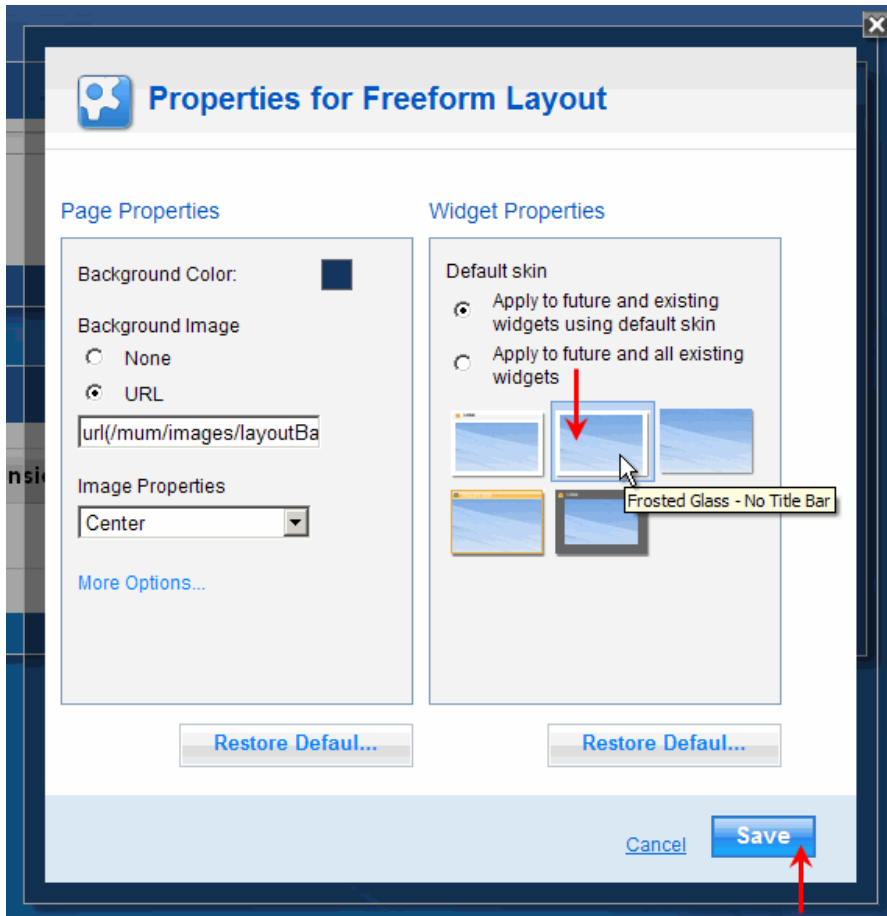


Figure 113. Selecting different skin

The “**Frosted Glass**” skin gives the application more streamlined appearance by removing title bars:

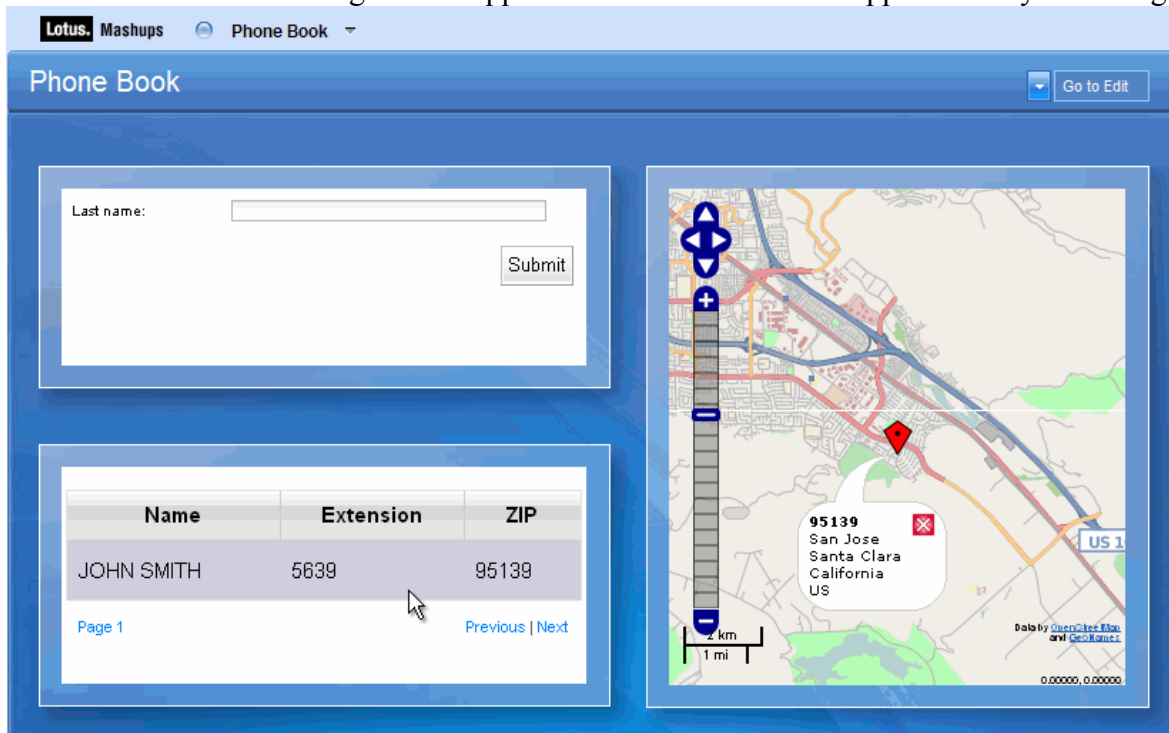


Figure 114. Final result after applying the “Frosted Glass” skin

Congratulations! You have completed the IMS Web 2.0 solution tutorial.

Resources

Learn

- Visit the [IMS Web 2.0 Solution Web site](#) and [IBM Mashup Center Web site](#) for more information and demo on IMS Web 2.0 solutions.
- To learn more about Web 2.0 and how businesses can gain value from Web 2.0 technologies, visit [Web 2.0 goes to work for business](#).
- For more overview information on Web 2.0 concepts and terminology and how IMS Web 2.0 solution fits in with the IMS On Demand solutions, see the [teleconference: IMS and Web 2.0 Go to Work](#).
- For more information on what RESTful services are and how to transform IMS assets into Web-oriented architecture, see the [teleconference: Web 2.0: Transforming IMS Enterprise SOA to Web Oriented Architecture](#).

Get products and technologies

- Download [trial versions of IBM Rational software](#).
- Download a trial version of [IBM Rational Developer for system z](#).

Generate an IMS feed using IMS Web 2.0 Solution for InfoSphere MashupHub

© Copyright IBM Corporation 2009. All rights reserved.

Discuss

- [Participate in the discussion forum.](#)
- Check out [developerWorks® blogs](#) and get involved in the [developerWorks community](#).

About the authors

- Evgueni Liakhovitch is a Software Developer at IBM.
- Yee-Rong Lai is an Information Developer at IBM.
- Shahin Mohammadi-Rashedi is an IMS SOA Demonstration Team Lead at IBM.

Trademark notice

IBM, the IBM logo and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.