

**Exploring Fabric-Aware Features of  
webMethods Developer 6.1 Feature Pack 1**

August 2004

Mark D. Carlson,  
Conneva, Inc.  
[mcarlson@conneva.com](mailto:mcarlson@conneva.com)

## Introduction

In early July 2004, webMethods released Feature Packs for Integration Server and Developer as well as a new version of Fabric that allow the user to perform the following functions from within Developer:

- Register Flow and Java Services as web services accessible to Fabric nodes
- Browse services available in the Fabric from within Developer
- Create web services connectors that bind to Fabric Services

This article will take a look at each of these features and then use the example services shipped with Fabric to demonstrate creation of a web services connector that invokes a web service and fails over automatically to an equivalent service with the original one is no longer available.

## Prerequisites

In order to explore these new features and to run the example, you will need the following:

- Integration Server 6.1 with Feature Pack 1
- Developer 6.1 with Feature Pack 1
- Fabric 1.0.2

You can obtain the Feature Packs using webMethods Installer. Authorized users can download webMethods Installer and Fabric 1.0.2 from the webMethods Advantage Software Downloads section.

When you install Fabric 1.0.2, accept the default options. This should change your classpath so that the `fabric-all.jar` and `fabric-examples.jar` libraries are at the beginning. On Windows, it will also add the `\fabric\bin` folder to your command path.

After the installation completes, you need to install the webMethods license file. The license file is usually provided in an email that is generated when you register for a Fabric evaluation copy or that is sent to your company after purchasing Fabric.

The easiest way to do install the license is to add the file to the `fabric-all.jar` file using the following steps:

- Shut down your Integration Server, if it is running.
- Copy your `webMethods-license.xml` file to the `\fabric\lib` folder
- Open a command prompt and navigate to the `\fabric\lib` folder
- Type `jar -uvf fabric-all.jar webMethods-license.xml`
- Restart your Integration Server

## Testing The Fabric 1.0.2 Installation

After restarting your workstation, confirm the success of your fabric installation by opening two command prompts and typing the following:

- Command prompt window #1: `java examples.fabric.Publish1`
- Command prompt window #2 `java examples.fabric.Invoke1`

If your installation was successful you should see results in the second command prompt window that look like this:

```
c:\>java examples.fabric.Invoke1
[STARTUP] Glue Professional 5.0.2 build 77 (c) 2001-2004 webMethods, Inc.
[STARTUP] Fabric 1.0.2 build 77 (c) 2001-2004 webMethods, Inc.
[STARTUP] subnet locator started on port 9030.
2 + 3 = 5
```

## Exploring Developer's New Fabric Features

After starting your Integration Server, you are almost ready launch Developer and explore the new Fabric-aware features provided in Feature Pack 1.

First you need to start a Fabric server. To do this type the following at a command prompt:

```
Fabric http://localhost:8006/examples
```

This will start a Fabric server on your local workstation (or server) that listens on port 8006. You should see something similar to the following:

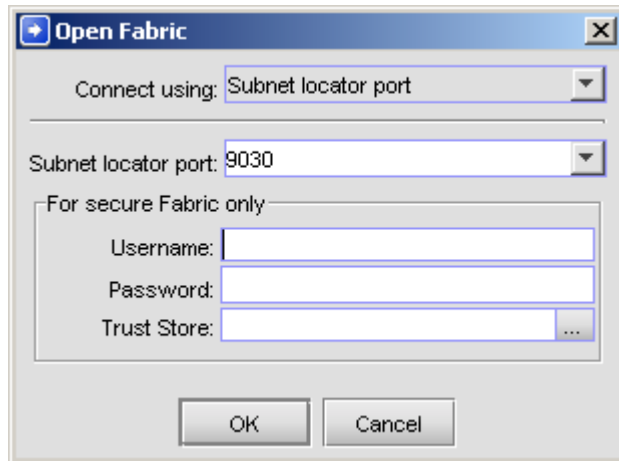
```
c:\>fabric http://localhost:8006/examples
[STARTUP] Glue Professional 5.0.2 build 77 (c) 2001-2004 webMethods, Inc.
[STARTUP] Fabric 1.0.2 build 77 (c) 2001-2004 webMethods, Inc.
[STARTUP] subnet locator started on port 9030.
[STARTUP] soap/http server started on http://155.157.209.140:8006/examples
```

Notice that Fabric is using Glue internally and that this instance of Fabric started something called a “subnet locator” on port 9030.

## Browsing Fabric using Developer

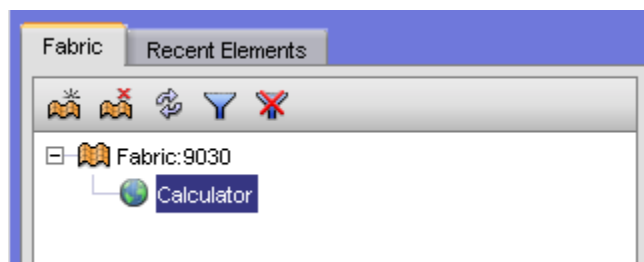
After launching Developer, the first thing to notice is the new Fabric browser window located in its own tab next to the “Recent Elements” tab in the bottom of Developer’s navigation panel.

Clicking on the only enabled button in this Fabric tab, will display the “Open Fabric” dialog. Note the default subnet locator port.



Click OK to use the default subnet locator port. You do not need to provide login credentials unless your Fabric is running in secure mode.

You should now see a list of service published to the Fabric that looks something like the following:



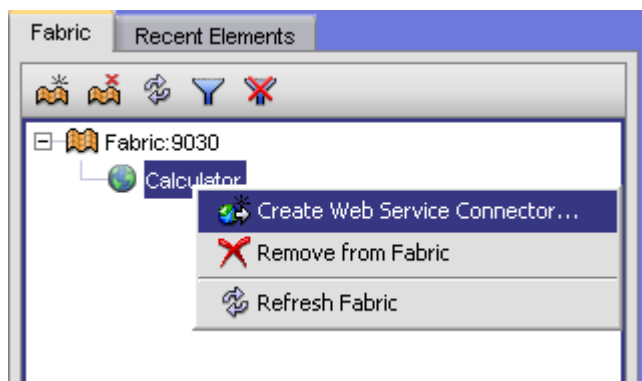
If you ran the Publish1 example used to confirm your installation, you should see a service named “Calculator” in the Fabric browser. Clicking on a service listed in the Fabric browser window will display that service’s properties in Developer’s properties panel. Most of these properties are read only at this time.

Properties	
Calculator	
Property	Value
[-] General	
name	Calculator
description	instance of class examples.fabric.Calculator ...
wSDL	http://155.157.209.140:8004/examples/shared/calculator.wsdl ...
endpoint	http://155.157.209.140:8004/examples/shared/calculator
since	7/27/2004 15:47:39 MDT
serviceKey	9B18AF37-C40C-5DC7-E975-23777B689939
custom	...
[-] Fabric internal	
online	true
messageSignature	2DA,AE438-0489-977E-74E6-DFFE091FA87A
interfaceSignature	3E7FA02E-285C-A8F3-2029-3C665D7E9215
database	
inFabricServer	true
systemService	

Notice the internal Fabric properties messageSignature and interfaceSignature. These properties are used to determine whether two web services are equivalent something which is necessary to implement failover. See the Fabric supplement to the Web Services Developer's Guide that ships with Developer Feature Pack 1 for more details on the remaining properties.

### Binding to Web Services in Fabric

When you bind to a web service in Fabric, Developer creates a web service connector. Right-clicking on the name of a service listed in the Fabric tab will bring up a context menu with two options 1) Create Web Service Connector and 2) Remove from Fabric.

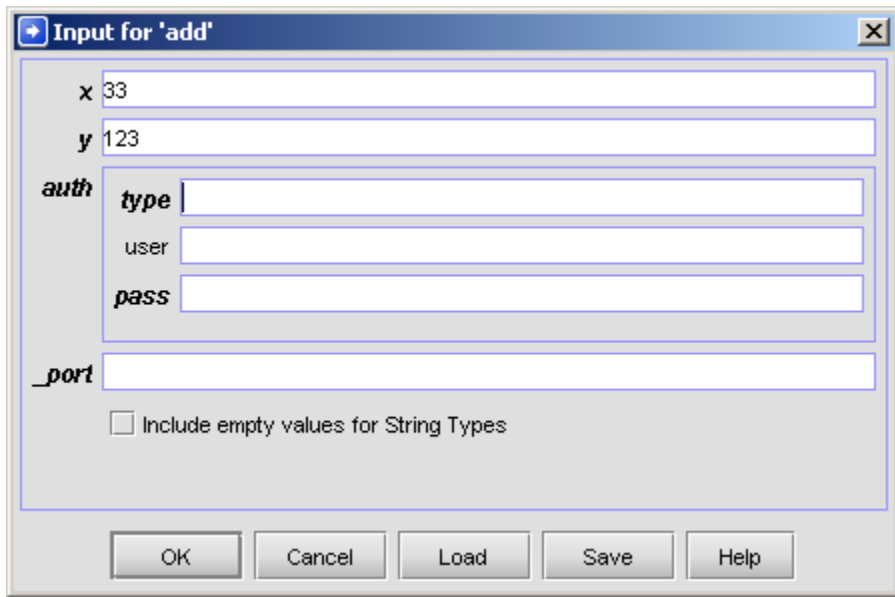


Choosing the “Create Web Service Connector” option will prompt you for a folder in which to create the web service connector. Clicking next will generate a Flow service and document types that will invoke the web service using either document/literal or soap RPC style as specified in the service’s WSDL file.




When you create a web service connector in this manner does not take advantage of Fabric’s location transparency or automatic failover features. To do that you must specify a special URL that requests a web service that matches certain criteria rather than one that binds to a particular instance. Location transparency and failover will be covered in a later section.

### Testing the Web Service Connector

Test the generated web service connector by double-clicking on the service in Developer’s navigation pane and using the Test->Run command. Specify the “x” and “y” input parameters. Because the Calculator service is running in unsecured mode you do not need to provide the “auth” parameters.



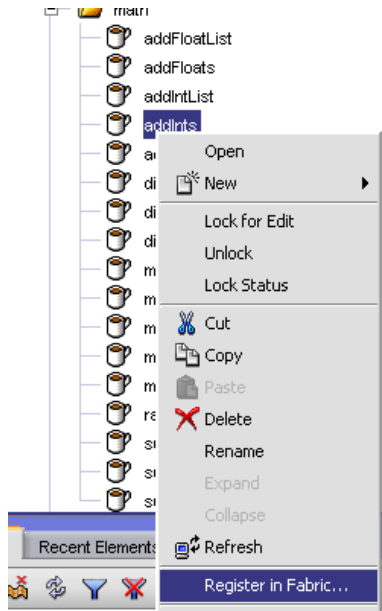
The results of the service will appear in Developer’s results pane as shown below. You should also see a line on the console window of the Publish1 service.

Results	
Name	Value
 x	33
 y	123
 addResult	156

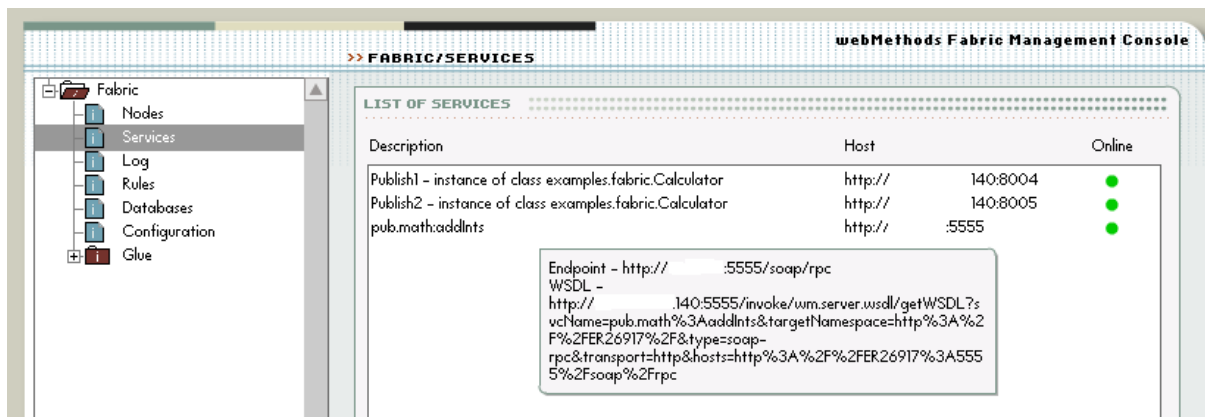
## Registering Integration Server Services in Fabric

Registering an Integration Server Flow or java service in Fabric makes that service available to other nodes in the Fabric.

To register an existing Flow or java service in Fabric, simply right-click on the name of the service (e.g. *pub.math.addints*) in Developer's navigation panel. The context menu that appears contains a new option called "Register in Fabric".



Choosing this option will display the WSDL Generator dialog box. To register the service using the Soap RPC style, accept the defaults. The service will be added to Fabric's registry and its WSDL file will be made available from the Fabric console.



## Testing Fabric's Location Transparency From webMethods Developer

In a press release in June 2004, webMethods introduced the concept of an "Enterprise Services Platform". One of the benefits identified in this announcement was location transparency for which the author gave the following definition:

*Location Transparency.* In a distributed system, the network takes care of discovering and finding where services live and then transparently invoking them when necessary. For enterprise assets and business services, the Enterprise Services Platform transparently manages the discovery and execution of services across the grid, eliminating the hardwiring between service consumers and producers that make systems less adaptable to change.

The default behavior of Integration Server web service connectors is endpoint addresses to be hardcoded into the generated Flow service. You can improve on this default approach by adding Flow statements to look up the endpoint from a properties file at runtime. However, achieving the location transparency in the definition above would take more work that most developers want to do.

With Developer Feature Pack 1, you can now generate web services connectors that will automatically discover available services that match a specified interface signature and message signature and automatically failover to an equivalent web service with the one located initially fails or is taken offline.

In this initial Feature Pack 1 version you need to specify a specially constructed URL when generating a web service connector. Hopefully, future Developer versions will improve on the ease of use in this area by providing a less cumbersome means accessing these key Fabric features.

## Step-By-Step Instructions for Testing Fabric's Location Transparency and Failover

1. Start a Fabric server by typing the following at the command prompt:  
`"fabric http://localhost:8006/examples"`
2. Start the first Calculator service by opening a command window and typing  
`"java examples.fabric.Publish1"`
3. Start the second Calculator service by opening another command window and typing  
`"java examples.fabric.Publish2"`
4. You should now see these two Calculator services in Developer's Fabric tab or in the Fabric console available at <http://localhost:8006/examples/console>.
5. Here's the ugly part. Write down the interface signature. In this release, you can't cut and paste it from either the Fabric console or the Developer properties panel. On the author's server the interfaceSignature is "3E7FA02E-285C-A8F3-2029-3C665D7E9215". If your interfaceSignature matches this one (and it should), you can cut and paste from this article.
6. In Developer select File-New and choose "Web Service Connector" from the dialog box. Choose the folder where you want the connector to be generated and

click “Next”. Choose “WSDL file” as the source of your WSDL then click “Next” again.

7. Here’s another tricky part. Enter the following in the WSDL location prompt:  
<http://localhost:8006/examples/wsdl?glue&service.interfaceSignature=3E7FA02E-285C-A8F3-2029-3C665D7E9215&failover&monitor>

Notice that the URL points to the Fabric server and not to one of the endpoints for a specific instance of the Calculator service. Fabric uses this URL to locate a web service that matches the specified interface signature and to return a WSDL file for that service that specifies a dynamic endpoint rather than an endpoint for a single Calculator service instance.



8. Run the resulting add or subtract web service connector in Developer using the Test->Run command. The console window of either the Publish1 or Publish2 service will display an output line showing the operation, operands and result of your invocation.
9. Close the command window that displayed the output by typing CTRL-C.
10. Run the service again. Fabric will detect the failure of the first service and fail over to a web service with the same messageSignature and interfaceSignature if one is available. The console output will appear on the second Calculator service window.
11. The command prompt console window of the Fabric server will display a message indicating that Fabric was initiating a failover to an equivalent service.

```
C:\webMethods6\fabric\lib>fabric http://localhost:8006/examples
[STARTUP] Glue Professional 5.0.2 build 77 (c) 2001-2004 webMethods, Inc.
[STARTUP] Fabric 1.0.2 build 77 (c) 2001-2004 webMethods, Inc.
[STARTUP] subnet locator started on port 9030.
[STARTUP] soap/http server started on http://localhost:8006/examples
[FAILOVER] rebind to ?glue&glue&service.interfaceSignature=3E7FA02E-285C-A8F3-
2029-3C665D7E9215&monitor&service.messageSignature=2DAAE438-0489-977E-74E6-
DFFE091FA87A&intermediary
```

## Summary

Since acquiring The Mind Electric last fall, webMethods has released new rebranded versions of Glue and Fabric and has demonstrated integration between Fabric and Integration Server at Integration World 2003.

The functionality provided by these new feature packs are the first visible evidence of Fabric's capabilities finding their way into the rest of the webMethods Integration Platform.

Developer's three new capabilities, discovering web services published to the fabric, registering IS web services with Fabric and binding to Fabric web services, are interesting previews of what the future holds for the rest of the product line.

The real value of these new feature packs goes beyond a technology preview, however. The ability to generate web services connectors that take advantage of Fabric's failover and load balancing capabilities addresses a real need of any organization attempting to build a robust web service-based service oriented architecture. This is harder today than it should be due to having to cobble together complex URL strings this should improve significantly in a future version of Developer.

## Resources

1. Fabric documentation is located in `<fabric_home>\docs\guides\index.html`
2. Documentation on the Fabric-aware features of Developer 6.1 Feature Pack 1 is contained in the "Web Services Developer Guide – Fabric Support Documentation Supplement – Version 6.1 FP1" document located in `<Developer_Home>\docs\guides`
3. Instructions on deploying a Fabric server inside the Integration Server's embedded Tomcat instance can be found in the article by Mark Carlson entitled "Configuring a Fabric Server to Run in the webMethods Integration Server's embedded Tomcat Instance" located on [www.wmusers.com](http://www.wmusers.com)

## About the Author

Mark D. Carlson is the founder and principal architect for Conneva, Inc. Conneva provides expert consulting services to webMethods customers in the areas of architecture design and development, technical account management services, mentoring and knowledge transfer. Mark began working with webMethods in 2000 while serving as a vice president in the IT organization of a large mortgage company. Mark is also the moderator of WM Users ([www.wmusers.com](http://www.wmusers.com)), an independent community of webMethods users. Mark can be reached by email at [mcarlson@conneva.com](mailto:mcarlson@conneva.com).