

Configuring a Fabric Server to Run in the Integration Server's Embedded Tomcat Instance

August 2004

Mark D. Carlson,
Conneva, Inc.
mcarlson@conneva.com

Background

Fabric nodes, operating in peer-to-peer fashion, provide several key Fabric services as “system” web services. These services including the endpoint manager, log manager, node manager, rule manager, service manager and the distributed XML database can be viewed from the Fabric console by clicking on the “Show System Services” checkbox in the “List of Service” window.

While Fabric's peer-to-peer architecture removes the requirement for Fabric nodes to run inside dedicated servers, good systems management practices will lead users to deploy Fabric's server nodes inside the managed servlet containers of application servers or dedicated servlet containers such as Apache Tomcat, Mort Bay Jetty, Coucho Resin or Macromedia JRun.

Because an “embedded” version of Tomcat has been included in Integration Server since version 6.01 another option is to deploy the Fabric web application there. Fabric's documentation provides information on the steps to follow to deploy Fabric inside IBM Websphere or BEA Weblogic. This “how-to” article will provide step-by-step instructions for creating a Fabric server node inside the Integration Server's embedded Tomcat instance.

Prerequisites

You will the following to be installed and operating correctly in order to successfully deploy a Fabric server inside Integration Server's Tomcat instance:

1. Integration Server 6.1 Feature Pack 1 or later (IS 6.01 may also work but was not tested for this article)
2. webMethods Developer 6.1 Feature Pack 1 or later (Developer 6.01 may also work)
3. Fabric 1.0.2 or later
4. A Fabric license file (usually named webMethods-license.xml)
5. Update access to the Integration Server's *\packages* folder
6. A user ID and password authorized to access the webMethods Administrator application

Step-By-Step Instructions

1. Using Developer, create a new package called FabricServer.
2. Create a new folder named “classes” as a subfolder of *<fabric_home>\webapps\fabric\WEB-INF* folder
3. Copy the Fabric license file “webMethods-license.xml” to the folder created in step 2
4. Replace the default web.xml file located in the WEB-INF folder with the one included in the appendix of this article
5. If your IntegrationServer is available at <http://localhost:5555>, proceed to Step XX

Configuring A Fabric Server to Run in the Integration Server's Tomcat Instance

6. Open the web.xml file in a text editor. Change the init-param value associated with the “electric.http.url” init-param with the hostname and port of your Integraton Server.

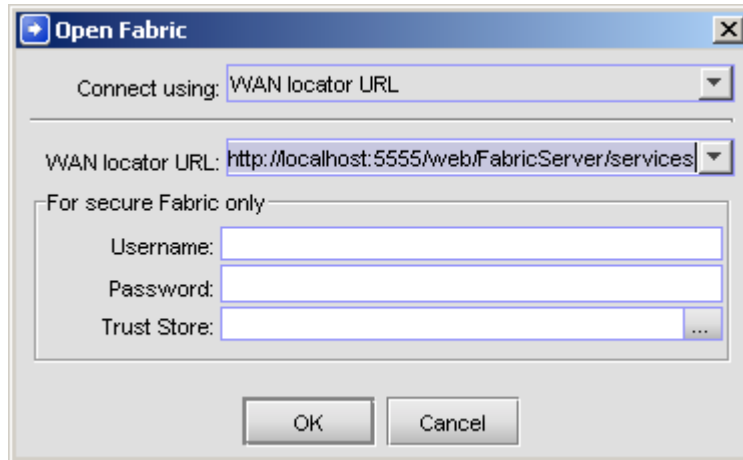
```
<init-param>
  <param-name>electric.http.url</param-name>
  <param-
value>http://<hostname>:<port>/web/FabricServer/services<
/param-value>
/init-param>
```

7. Copy the `<fabric_home>\webapps\fabric\WEB-INF` folder to your Integration Server's `\packages\FabricServer\web` folder
8. Open the webMethods Administrator tool and reload the FabricServer package by clicking on the reload icon in the Packages->Management page. When this completes you should see log messages similar to the following in the Integration Server's server.log file

```
[3148]2004-08-10 11:42:07 MDT [ISS.0028.0033V1] Package FabricServer has reloaded.
[3147]2004-08-10 11:42:07 MDT [ISS.0100.0008V1] StandardWrapper[/web/FabricServer:invoker]:
Loading container servlet invoker
[3146]2004-08-10 11:41:54 MDT [ISS.0100.0008V1] ContextConfig[/web/FabricServer]: Added
certificates -> request attribute Valve
[3145]2004-08-10 11:41:54 MDT [ISS.0100.0008V1] StandardManager[/web/FabricServer]: Seeding of
random number generator has been completed
[3144]2004-08-10 11:41:54 MDT [ISS.0100.0008V1] StandardManager[/web/FabricServer]: Seeding
random number generator class java.security.SecureRandom
[3143]2004-08-10 11:41:54 MDT [ISS.0100.0008V1] WebappLoader[/web/FabricServer]: Deploy JAR
/WEB-INF/lib/fabric.jar to C:\webMethods6\IntegrationServer\.\packages\FabricServer\web\WEB-
INF\lib\fabric.jar
[3142]2004-08-10 11:41:54 MDT [ISS.0100.0008V1] WebappLoader[/web/FabricServer]: Deploy JAR
/WEB-INF/lib/dom.jar to C:\webMethods6\IntegrationServer\.\packages\FabricServer\web\WEB-
INF\lib\dom.jar
[3141]2004-08-10 11:41:54 MDT [ISS.0100.0008V1] WebappLoader[/web/FabricServer]: Deploying
```

9. Access the console of the embedded Fabric server using the following URL:
<http://localhost:5555/web/FabricServer/console>
10. Developer 6.1 with Feature Pack 1 can browse Fabric services, publish IS Flow and java services to the Fabric and bind to Fabric services. To have Developer join the embedded Fabric server use the Session->Open Fabric menu command, choose “WAN locator URL” from the “Connect Using drop down list box and specify the following for the “WAN locator URL”
<http://localhost:5555/web/FabricServer/services>

Configuring A Fabric Server to Run in the Integration Server's Tomcat Instance



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 web applications to the Integration Server's embedded Tomcat instance can be found in the “Java Server Pages Developer's Guide” located in `<Developer_Home>\docs\guides`
4. Step-by-step instructions for exploring Fabric's discovery, binding and publishing capabilities using webMethods Developer 6.1 Feature Pack 1 can be found in the article by Mark Carlson entitled “Exploring Fabric-Aware Feature of webMethods Developer 6.1 Feature Pack 1” located on 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), an independent community of webMethods users. Mark can be reached by email at mcarlson@conneva.com.

Appendix A – Contents of “web.xml” file for the Embedded Fabric Server

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
'http://java.sun.com/j2ee/dtds/web-app_2.2.dtd'>
```

```
<web-app>
  <servlet>
    <servlet-name>glue-soap</servlet-name>
    <servlet-class>electric.server.http.ServletServer</servlet-class>

    <init-param>
      <param-name>httpRegistryRoot</param-name>
      <param-value></param-value>
    </init-param>
```

```
    <init-param>
      <param-name>electric.http.url</param-name>
```

```
<!--
```

Note: Change hostname and port in the param value below to point to the Integration Server in which a Fabric server node has been deployed to the embedded Tomcat instance. Replace "FabricServer" with the name of the IS package where fabric.war or \fabric\webapps\fabric\WEB-INF folder has been copied to the package's \web folder

After the embedded server has been successfully configured, the Fabric console can be accessed at `http://<IS_hostname>:<port>/web/FabricServer/console`.

Developer 6.1 Feature Pack 1 and later can access the embedded Fabric server by choosing "WAN locator URL" from the "Connect Using" list box and specifying `http://<IS_hostname>:<port>/web/FabricServer/services` as the "WAN locator URL"

```
-->
```

```
  <param-value>http://localhost:5555/web/FabricServer/services</param-value>
</init-param>
```

```
  <load-on-startup>1</load-on-startup>
</servlet>
```

```
<servlet>
  <servlet-name>glue-console</servlet-name>
  <servlet-class>electric.console.ConsoleServlet</servlet-class>
```

```
  <init-param>
    <param-name>soapServletContext</param-name>
    <param-value>/services</param-value>
  </init-param>
```

```
  <load-on-startup>2</load-on-startup>
</servlet>
```

```
<servlet>
  <servlet-name>fabric-wsdl</servlet-name>
  <servlet-class>com.webmethods.fabric.intermediary.WSDLHandler</servlet-class>
```

```
  <load-on-startup>3</load-on-startup>
</servlet>
```

```
<servlet-mapping>
  <servlet-name>glue-soap</servlet-name>
  <url-pattern>/services/*</url-pattern>
</servlet-mapping>
```

```
<servlet-mapping>
  <servlet-name>glue-console</servlet-name>
  <url-pattern>/console/*</url-pattern>
</servlet-mapping>
```

```
<servlet-mapping>
  <servlet-name>fabric-wsdl</servlet-name>
  <url-pattern>/wsdl/*</url-pattern>
</servlet-mapping>
```

```
</web-app>
```