

# Web Services in Eclipse

Lon Riesberg

CSCI 7818 – Web Services

September 17, 2008



LABORATORY FOR ATMOSPHERIC AND SPACE PHYSICS



September 17, 2008 | HOME | SCIENCE | EDUCATION | ENGINEERING | MISSION OPS | PERSONNEL

## WELCOME TO LASP

### FEATURE

May 21 2008

#### Getting scientific with Google Earth



As part of Google's ongoing mission to make information more readily available we recently collaborated with the University of Colorado, Boulder to hold an event for scientists and researchers in the area. The event focused on finding ways to use KML and Google Earth to display and communicate scientific research with policy makers, students, and the public.

The event was hosted by the Laboratory for Atmospheric and Space Physics as part of the Electronic Geophysical Year program. [Learn More](#)

#### Saturn's Auroras

UVIS obtained 26 images of Saturn's auroras on May 25, 2007 over an 8 hour 15 minute period, most of a planetary rotation period. They are presented here as side-by-side animations of simultaneous observations from two UVIS channels, the Extreme Ultraviolet (EUV) and the Far Ultraviolet (FUV). [Learn more / view video](#)



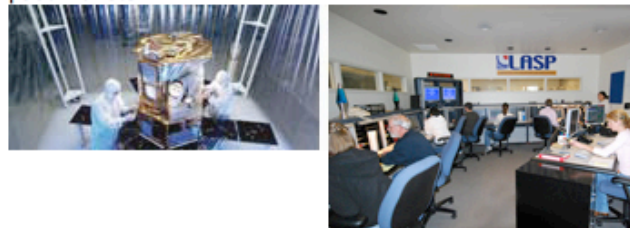
### SCIENCE

[Solar Influences](#) | [Atmospheric Science](#) | [Space Physics](#) | [Planetary Physics](#) | [Center for Astrobiology](#)



### EDUCATION & OUTREACH

[Undergraduate](#) | [Graduate/Ph.D.](#) | [K - 12 Students & Teachers](#) | [Journalists](#)

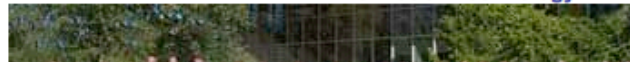


### ENGINEERING

[Technical Capabilities](#) | [Calibration & Test](#) | [Quality Assurance](#)

### MISSION OPS

[Flight Operations](#) | [Planning and Scheduling](#) | [Software Tools](#) | [Data Systems](#) | [Information Technology](#)



### ABOUT LASP

LASP was born in 1948 as the Upper Atmosphere Lab (UAL) along with a handful of other American universities and the military to initiate the era of space exploration...[\(more\)](#)

- [Employment](#)
- [Seminars](#)
- [Public Lectures](#)
- [Visitor Info / Maps & Directions](#)
- [LASP Mission Statement](#)
- [Mission/Project History](#)
- [LASP Activity Report 2006, 2007](#)

### PRODUCTS

- [LISIRD \(Solar Irradiance Datacenter\)](#)
- [LASP Data Products](#)
- [LASP Software Tools](#)
- [Publications](#)
- [IGY Legacy Videos](#)

### LASP IN THE NEWS

**NASA SELECTS CU-Boulder TO LEAD \$485 MILLION MARS MISSION**



Sept. 15. In the largest research contract

# Outline

- 🌐 *Eclipse Web Tools Platform Project*
  - 🌐 *What it can do*
  - 🌐 *High Level Architecture*
  - 🌐 *System Requirements*
  - 🌐 *Demos*
    - 🌐 *Top-down Development*
    - 🌐 *Bottom-up Development*
  - 🌐 *Resources*






# Eclipse WTP - Features

-  Standard Web Tools
-  Wizard that guides the user through the generate/deploy/test/publish lifecycle of a web service
-  Supports bottom-up (from Java technology) and top-down (from WSDL) web service creation
-  Configures project, server, and SOAP engine
-  Code generators
-  Test facilities




# Eclipse WTP – System Req's





## *What I needed...*

-  *Eclipse*
-  *Java Development Kit*
-  *Eclipse WebTools*
-  *Tomcat*
-  *Axis 2*

## *AND...*

-  *Sysdeo Tomcat Launcher – offers easy access to Tomcat configuration. Launching web services would not work for me on Mac OS X without this plug-in.*

# Creating a Web Service

-  Eclipse WTP offers wizards that guide the user through the generate/deploy/test/publish lifecycle of a Web service
-  Configures project, server, and SOAP engine
-  Code generators
-  Test facilities

# Creating a Web Service – Bottom-up

- 🌐 Write Java class that implements server functionality.
- 🌐 Create a web project:  
File -> New -> Other -> Dynamic Web Project (select local Tomcat server as target)
- 🌐 Import Java class into new web project.
- 🌐 Create web service and client:  
right-click Java class -> New -> Other -> Web Service



# Create Web Service From Java Class

The screenshot displays the Eclipse IDE interface during the creation of a web service. The main editor shows the following Java code in `Converter.java`:

```
package wtp;

public class Converter
{
    public float celsiusToFahrenheit ( float celsius )
    {
        return (celsius * 9 / 5) + 32;
    }

    public float fahrenheitToCelsius ( float fahrenheit )
    {
        return (fahrenheit - 32) * 5 / 9;
    }
}
```

The **Web Service** wizard is open, showing the following configuration:

- Web Services:** Select a service implementation or definition and move the sliders to set the level of service and client generation.
- Web service type:** Bottom up Java bean Web Service
- Service implementation:** wtp.Converter
- Test service:** Configuration: Server: Tomcat v6.0 Server, Web service runtime: Apache Axis, Service project: Converter
- Client type:** Java Proxy
- No client:** Configuration: No client generation.
- Publish the Web service
- Monitor the Web service
- Overwrite files without warning

The **Servers** view shows the **Tomcat v6.0 Server at localhost** is **Started**.

# Resulting WSDL

The screenshot shows the Eclipse IDE interface. The main editor displays the following WSDL content:

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://wtp" xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:impl="http://wtp" xmlns:intf="http://wtp" xmlns:wsdl="http://wsdl.org/2003/03/01" >
  <!--WSDL created by Apache Axis version: 1.4
  Built on Apr 22, 2006 (06:55:48 PDT)-->
  <wsdl:types>
    <schema elementFormDefault="qualified" targetNamespace="http://wtp" xmlns="http://www.w3.org/2001/XMLSchema">
      <element name="celsiusToFahrenheit">
        <complexType>
          <sequence>
            <element name="celsius" type="xsd:float"/>
          </sequence>
        </complexType>
      </element>
      <element name="celsiusToFahrenheitResponse">
        <complexType>
          <sequence>
            <element name="celsiusToFahrenheitReturn" type="xsd:float"/>
          </sequence>
        </complexType>
      </element>
      <element name="fahrenheitToCelsius">
        <complexType>
          <sequence>
            <element name="fahrenheit" type="xsd:float"/>
          </sequence>
        </complexType>
      </element>
      <element name="fahrenheitToCelsiusResponse">
        <complexType>
          <sequence>
            <element name="fahrenheitToCelsiusReturn" type="xsd:float"/>
          </sequence>
        </complexType>
      </element>
    </schema>
  </wsdl:types>

```

The interface also shows a Navigator on the left with a project structure including Converter.java and Converter.wsdl. At the bottom, the Servers view shows Tomcat v6.0 Server at localhost in a Started state.

Server	State	Status
Tomcat v6.0 Server at localhost	Started	Restart

# WSDL in Design Editor

The screenshot shows the Eclipse IDE interface with the WSDL in Design Editor. The main window displays a diagram of the ConverterService and Converter. ConverterService is a port type with a binding to Converter. Converter has two operations: celsiusToFahrenheit and fahrenheitToCelsius. The Outline view on the right shows the project structure, including ConverterService and Converter. The Servers view at the bottom shows Tomcat v6.0 Server at localhost.

**ConverterService**

- Converter (http://localhost:8080/...)

**Converter**

Operation	Input	Output
celsiusToFahrenheit	parameters	celsiusToFahrenheitResponse
fahrenheitToCelsius	parameters	fahrenheitToCelsiusResponse

**Outline**

- Imports
  - Types
    - http://wtp
- Services
  - ConverterService
    - Converter
- Bindings
  - ConverterSoapBinding
- Port Type
  - Converter
- Messages
  - fahrenheitToCelsiusResponse
    - parameters
  - celsiusToFahrenheitResponse
    - parameters
  - fahrenheitToCelsiusRequest
    - parameters
  - celsiusToFahrenheitRequest
    - parameters

**Servers**

Server	State	Status
Tomcat v6.0 Server at localhost	Started	Restart







# Running...

The screenshot displays the Eclipse IDE interface for a Java EE project named "Converter". The main workspace is divided into several views:

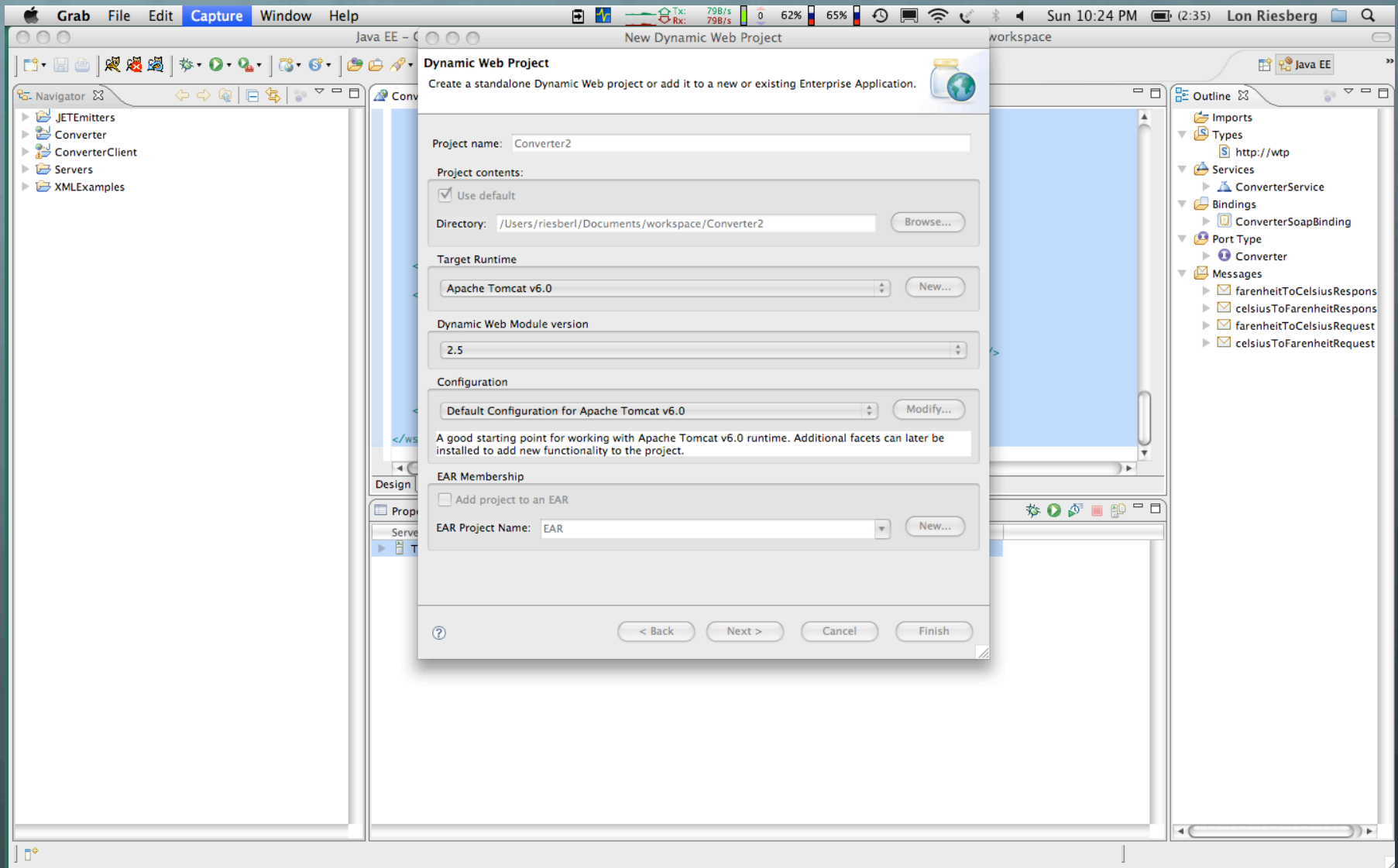
- Navigator:** Shows the project structure, including folders like ".settings", "build", "src", "wtp", "WebContent", and "wsdl". The "ConverterService" package is expanded, showing the "fahrenheitToCelsius" operation.
- Web Services Explorer:** This view is active and shows the "Invoke a WSDL Operation" panel. It displays the endpoint "http://localhost:8080/Converter/services/Converter" and the operation "fahrenheitToCelsius". The parameter "fahrenheit" is set to "212". The "Status" panel shows the response "fahrenheitToCelsiusReturn (float): 100.0".
- Servers:** Shows the "Tomcat v6.0 Server at localhost" with a "Started" status.

The top of the IDE shows the menu bar (Grab, File, Edit, Capture, Window, Help) and the status bar (Sun 3:58 PM, Lon Riesberg). The browser address bar shows "Java EE - http://127.0.0.1:50276/wse/wsexplorer/wsexplorer.jsp?org.eclipse.wst.ws.explorer=1 - Eclipse Platform - /Users/riesberl/Documents/workspace".

# Creating a Web Service – Top-down

-  Create WSDL document
-  Create a web project:  
File -> New -> Other -> Dynamic Web Project (select local Tomcat server as target)
-  Create a corresponding web client:  
right-click WSDL document -> File -> New -> Other -> Web Services -> Web Client
-  Flesh out server functionality in resultant Java stub.

# Create Web Project From WSDL





# Create Web Service From WSDL

The screenshot displays the Eclipse IDE interface with the 'Web Services' wizard open. The wizard is configured to create a 'Top down Java bean Web Service' from the 'Converter.wsdl' file. The 'Client type' is set to 'Java Proxy'. The configuration includes a 'Start service' diagram and a 'No client' diagram. The 'Overwrite files without warning' checkbox is checked. The background shows the project structure in the Navigator, the WSDL code in the Editor, and the Outline view.

**Web Services**  
Select a service implementation or definition and move the sliders to set the level of service and client generation.

Web service type: **Top down Java bean Web Service**

Service definition: **/Converter/WebContent/wsdl/Converter.wsdl**

**Start service**  
Configuration:  
[Server: Tomcat v6.0 Server](#)  
[Web service runtime: Apache Axis](#)  
[Service project: Converter2](#)

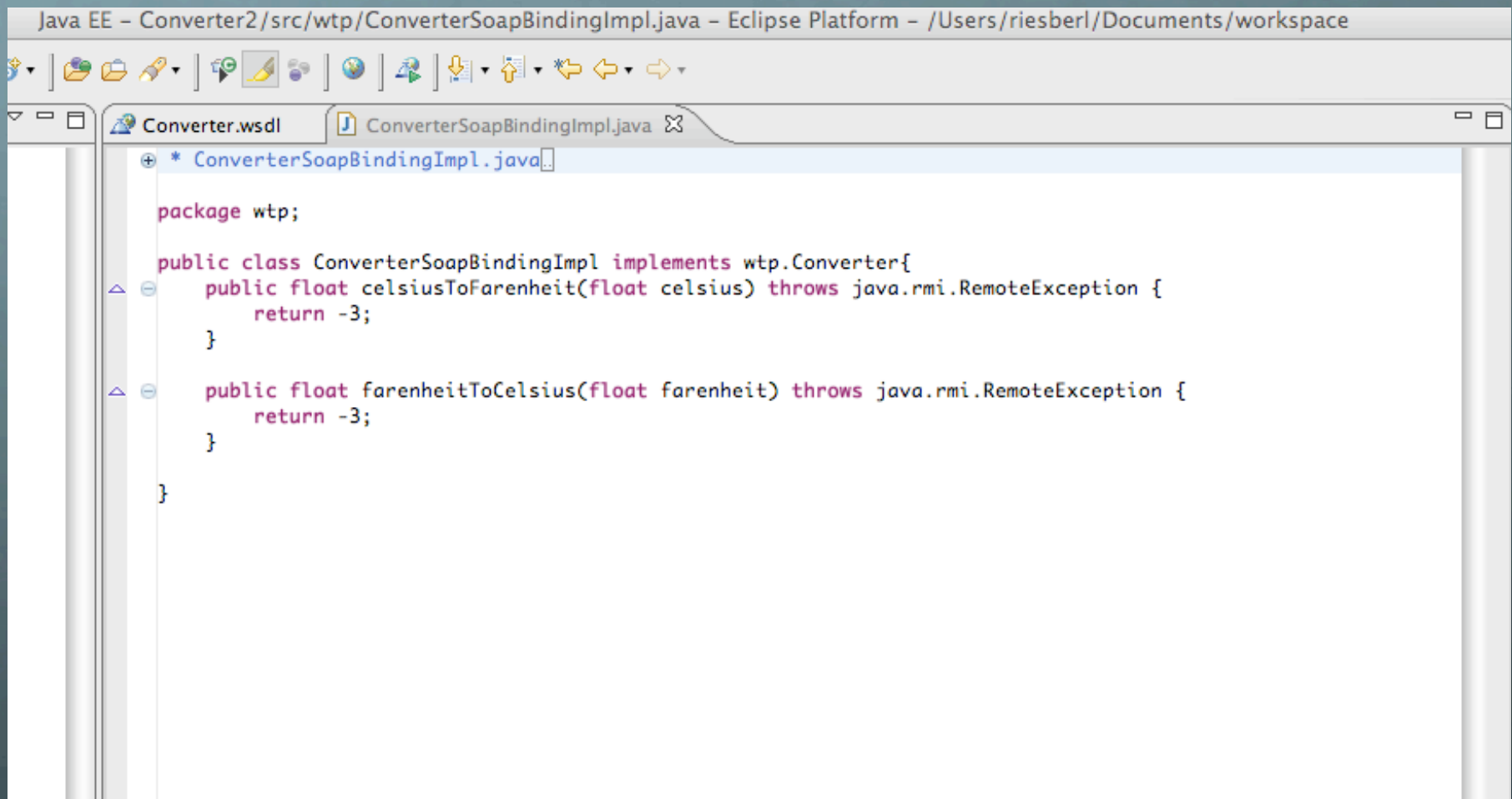
**Client type: Java Proxy**

**No client**  
Configuration: No client generation.

Publish the Web service  
 Monitor the Web service  
 Overwrite files without warning

```
</wsdl:input>
<wsdl:output name="farenh...
<wsdlsoap:body use="li...
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="ConverterSe...
<wsdl:port binding="impl:Con...
<wsdlsoap:address location...
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

# Flesh out Resulting Java Stubs



```
Java EE - Converter2/src/wtp/ConverterSoapBindingImpl.java - Eclipse Platform - /Users/riesberl/Documents/workspace

Converter.wsd | ConverterSoapBindingImpl.java

* ConverterSoapBindingImpl.java

package wtp;

public class ConverterSoapBindingImpl implements wtp.Converter{
    public float celsiusToFahrenheit(float celsius) throws java.rmi.RemoteException {
        return -3;
    }

    public float fahrenheitToCelsius(float fahrenheit) throws java.rmi.RemoteException {
        return -3;
    }
}
```

# Running...

The screenshot displays the Eclipse IDE interface with the following components:

- Navigator:** Shows a project tree for 'Converter2' with subfolders for '.settings', 'build', and 'src'. The 'src' folder contains several Java files including 'Converter.java', 'ConverterService.java', 'ConverterServiceLocator.java', 'ConverterSoapBindingImpl.java', 'ConverterSoapBindingSkeleton.java', and 'ConverterSoapBindingStub.java'. Other folders include 'WebContent', '.classpath', '.project', 'ConverterClient', 'Servers', and 'XMLExamples'.
- Web Services Explorer:** Contains a 'Navigator' pane showing a WSDL structure with 'ConverterService' and 'ConverterSoapBinding' elements. The 'Actions' pane is active, showing 'Invoke a WSDL Operation' with a 'Source' link. The 'Endpoints' field is set to 'http://localhost:8080/Converter2/services/Converter'. The 'Body' section is expanded to show the 'fahrenheitToCelsius' operation with a 'fahrenheit' parameter set to '212'. Below this, the 'Status' pane shows the response body: 'fahrenheitToCelsiusResponse' with 'fahrenheitToCelsiusReturn (float): 100.0'.
- Servers View:** Located at the bottom, it shows a table with the following data:

Server	State	Status
Tomcat v6.0 Server at localhost	Started	Restart



# Eclipse WTP - Resources

-  *Web Tools Platform - <http://eclipse.org/webtools>*
-  *Web Tools Community*
-  *Eclipse Built-in Help*
-  *Google – “eclipse web services tutorial”*