

**CS 683 Emerging Technologies**  
**Fall Semester, 2004**  
**Doc 10 Seaside & Cocoon Hello World**  
**Contents**

References.....	2
Reading.....	2
Seaside Hello World Example.....	3
Registering Web Application via Web Interface.....	5
Seaside Bug and Registering Web Apps.....	7
Cocoon Hello World Example.....	9
Simple Example.....	9
Hello World with CSS.....	13
Hello World with Java Date.....	16
Hello World with JavaScript Date.....	18
Hello World & Java Date in Separate Files.....	20

Copyright ©, All rights reserved. 2004 SDSU & Roger Whitney,  
5500 Campanile Drive, San Diego, CA 92182-7700 USA.  
OpenContent (<http://www.opencontent.org/opl.shtml>) license  
defines the copyright on this document.

## References

Cocoon Developer's Handbook, Moczar & Aston, Sams Publishing, 2003

Cocoon User Documentation,  
<http://cocoon.apache.org/2.1/userdocs/index.html>

Seaside Source Code

## Reading

Cocoon Developer's Handbook, Moczar & Aston, Sams Publishing, 2003 Chapters 7 & 8

Seaside Tutorial  
<http://www.cincomsmalltalk.com/CincomSmalltalkWiki/Seaside%20Tutorial>

## Seaside Hello World Example

```
Smalltalk.Seaside defineClass: #HelloWorld
  superclass: #{Seaside.WAComponent}
  indexedType: #none
  private: false
  instanceVariableNames: "
  classInstanceVariableNames: "
  imports: "
  category: 'CS683'
```

### Instance Methods

```
renderContentOn: html
  html heading: 'Hello World' level: 2.
  html paragraph: 'Hi Mom. Today is ' ,
    TimeStamp now printString
```

```
style
  ^'h2
  { color: red;
  text-align: center;
  font-size: xx-large;}
```

```
p { text-align: left; }'
```

```
updateRoot: anHtmlRoot
  super updateRoot: anHtmlRoot.
  anHtmlRoot title: 'Hello World'
```

## Class Methods

```
canBeRoot
```

```
  ^ true
```

```
initialize
```

```
  "self initialize"
```

```
  (self registerAsApplication: 'helloWorld')
```

canBeRoot indicates that the class can act as a top level web application. The default value is false, which means the class can only be used a component of a top level web application.

The initialize method shows how to have the class register it self as a top level web application. The argument to registerAsApplication: give the end of the url to access web application. Using the default setting of seaside this web application can be accessed as:

```
http://127.0.0.1:8008/seaside/go/helloWorld
```

There is a Web interface that allows you to set a number of parameters. It is the way I normally register a web app in seaside.

## Registering Web Application via Web Interface

Assuming the default Seaside settings go to

<http://127.0.0.1:8008/seaside/go/config>

Enter the name and password you created when loading Seaside. In the middle of the resulting page you will the form:

**add entry point:**

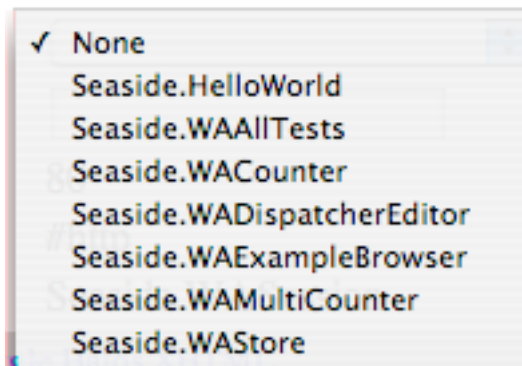
Path:

Type:

Enter the name for the web app that users will enter in the url, say myHelloWorld and click on add. In the middle of the configure application page you will see the popup menu:

Main Class	Seaside.WARenderLoopMain	<a href="#">override</a>
Root Component	<input type="text" value="None"/>	(unspecified)
Server Hostname	<input type="text"/>	(unspecified)

Click on the Root Component popup menu. It will contain a list of all the classes that can be a root or a top level web application.



Select Seaside.HelloWorld. At the bottom of the page there are two buttons:



First click on “Save” then click on “Done”. After you click on the “Done” button you will go back the main configuration page. At the top of the page you will see a list of the registered web applications for the site:

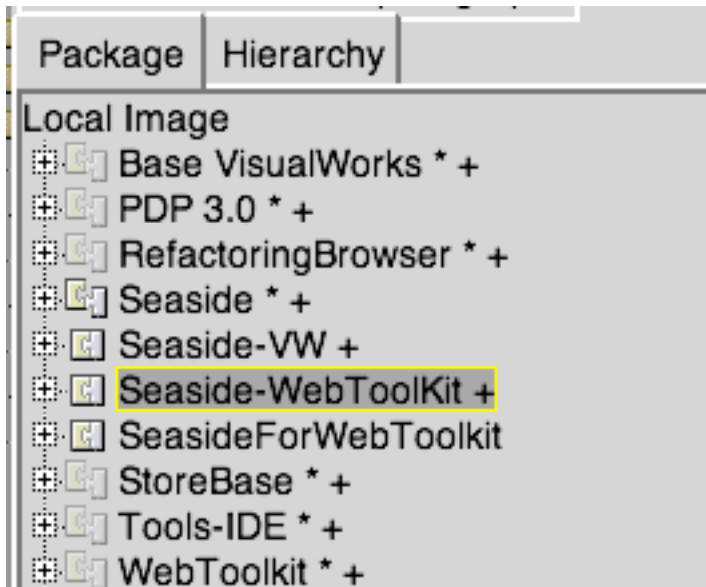
**/seaside/go**

<a href="#">config</a>	<a href="#">configure</a> <a href="#">remove</a>
<a href="#">counter</a>	<a href="#">configure</a> <a href="#">remove</a>
<a href="#">countFoo</a>	<a href="#">configure</a> <a href="#">remove</a>
<a href="#">multi</a>	<a href="#">configure</a> <a href="#">remove</a>
<a href="#">myHelloWorld</a>	<a href="#">configure</a> <a href="#">remove</a>
<a href="#">store</a>	<a href="#">configure</a> <a href="#">remove</a>

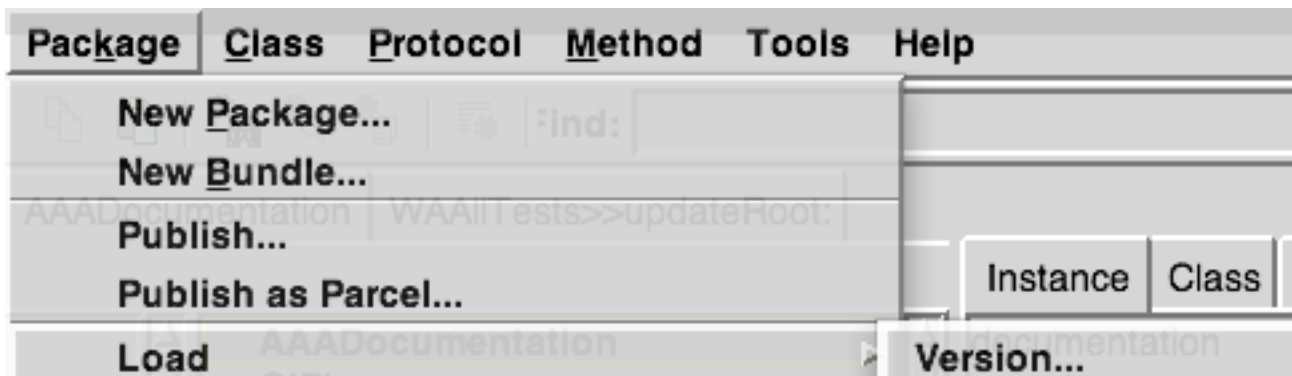
Click on myHelloWorld to go to the web app. The configure link after myHelloWorld brings you back the configuration page for the web app. The remove link removes the web app from the list of registered web apps for the site.

## Seaside Bug and Registering Web Apps

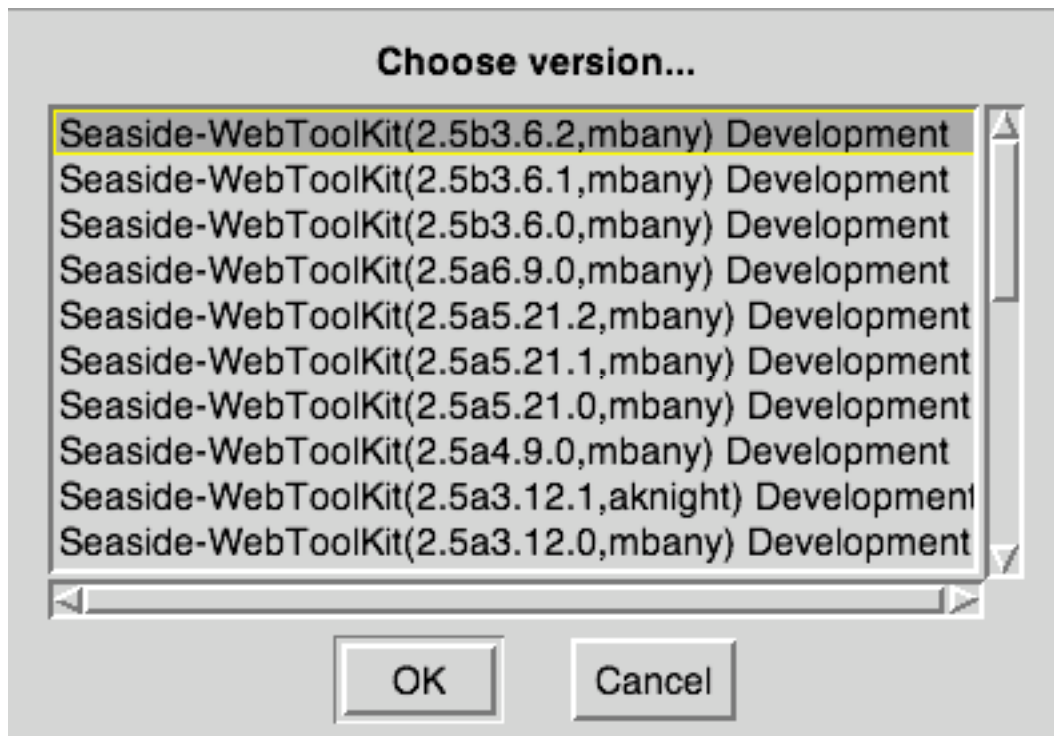
Prior to Sept 20, 2004 Seaside in VW had a bug that affected registering Web Apps. To load the fix first connect to the Cincom repository. Then VW System browser select the Seaside-WebToolKit package as below.



Now in the “Package” menu select “Load” item and the “Version...” sub item.



You will see a list of versions of the package. Select the version 2.5b3.6.2,mbany. Then click ok. Once the package is loaded save your image. The bug is now fixed.



## Cocoon Hello World Example Simple Example

CATALINA\_HOME = location of Tomcat

In CATALINA\_HOME/webapps/cocoon create directories:

hello  
hello/content  
hello/style  
hello/resources

### **hello/content/main.xml**

```
<?xml version="1.0"?>
<page>
  <title>Hello World</title>
  <content>
    <paragraph>Hi Mom</paragraph>
  </content>
</page>
```

**hello/style/main.xsl**

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="page">
    <html>
      <head>
        <title> <xsl:value-of select="title"/></title>
      </head>
      <body>
        <xsl:apply-templates/>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="title">
    <h2><xsl:apply-templates/></h2>
  </xsl:template>

  <xsl:template match="paragraph">
    <p><xsl:apply-templates/></p>
  </xsl:template>
</xsl:stylesheet>
```

## Tying Content & XSLT together

### hello/sitemap.xmap

```
<?xml version="1.0"?>

<map:sitemap xmlns:map="http://apache.org/cocoon/sitemap/1.0">
  <map:components>
    <map:generators default="file"/>
    <map:transformers default="xslt"/>
    <map:readers default="resources"/>
    <map:serializers default="html"/>
    <map:selectors default="browser"/>
    <map:matchers default="wildcard"/>
  </map:components>

  <map:pipelines>
    <map:pipeline>
      <map:match pattern="index.html">
        <map:generate type="file" src="content/main.xml"/>
        <map:transform type="xslt" src="style/main.xsl"/>
        <map:serialize type="html"/>
      </map:match>
    </map:pipeline>
  </map:pipelines>
</map:sitemap>
```

## Registering the Web App

### CATALINA\_HOME/webapps/cocoon/sitemap.xmap

```
<map:pipeline>
  <map:match pattern="cs683/**">
    <map:mount uri-prefix="cs683" src="hello/sitemap.xmap"
      check-reload="yes" reload-method="synchron"/>
  </map:match>
</map:pipeline>
```

Add the above inside the `<map:pipelines>` `</map:pipelines>` of this file. Do not change any else in the file

Example url

<http://machineName:TomcatPort/cocoon/cs683/index.html>

## Hello World with CSS

### New File

hello/resources/cs683.css

### Changed Files

hello/style/main.xsl

hello/sitemap.xmap

### **hello/resources/cs683.css**

h2

```
{  
  color: red;  
  text-align: center;  
  font-size: xx-large;  
}
```

```
p { text-align: left; }
```

**hello/style/main.xsl**

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="page">
    <html>
      <head>
        <title><xsl:value-of select="title"/> </title>
        <link rel="Stylesheet" type="text/css" media="screen"
          href="cs683.css" />
      </head>
      <body><xsl:apply-templates/></body>
    </html>
  </xsl:template>

  <xsl:template match="title">
    <h2><xsl:apply-templates/></h2>
  </xsl:template>

  <xsl:template match="paragraph">
    <p><xsl:apply-templates/></p>
  </xsl:template>

</xsl:stylesheet>
```

## hello/sitemap.xmap

```
<?xml version="1.0"?>
<map:sitemap xmlns:map="http://apache.org/cocoon/sitemap/1.0">
  <map:components>
    <map:generators default="file" />
    <map:transformers default="xslt" />
    <map:readers default="resources" />
    <map:serializers default="html" />
    <map:selectors default="browser" />
    <map:matchers default="wildcard" />
  </map:components>
  <map:pipelines>
    <map:pipeline>
      <map:match pattern="index.html">
        <map:generate type="file" src="content/main.xml" />
        <map:transform type="xslt" src="style/main.xsl" />
        <map:serialize type="html" />
      </map:match>
      <map:match pattern="*.css">
        <map:read type="resource" src="resources/{1}.css"
          mime-type="text/css">
        </map:read>
      </map:match>
    </map:pipeline>
  </map:pipelines>
</map:sitemap>
```

## Hello World with Java Date

### New File

hello/content/main.xsp

### Modified File

hello/sitemap.xmap

### hello/content/main.xsp

```
<?xml version="1.0"?>
<xsp:page language="java" xmlns:xsp="http://apache.org/xsp">

<xsp:structure>
  <xsp:include>java.util.Date</xsp:include>
</xsp:structure>

<xsp:logic>
  Date now = new Date();
</xsp:logic>

<page>
  <title>Hello World</title>
  <content>
    <paragraph>Hi Mom. Today is <xsp:expr>now</xsp:expr>
  </paragraph>
  </content>
</page>
</xsp:page>
```

## hello/sitemap.xmap

```
<?xml version="1.0"?>
<map:sitemap xmlns:map="http://apache.org/cocoon/sitemap/1.0">
  <map:components>
    <map:generators default="file" />
    <map:transformers default="xslt" />
    <map:readers default="resources" />
    <map:serializers default="html" />
    <map:selectors default="browser" />
    <map:matchers default="wildcard" />
  </map:components>
  <map:pipelines>
    <map:pipeline>
      <map:match pattern="index.xsp">
        <map:generate type="serverpages" src="content/main.xsp" />
        <map:transform type="xslt" src="style/main.xsl" />
        <map:serialize type="html" />
      </map:match>
      <map:match pattern="*.css">
        <map:read type="resource" src="resources/{1}.css"
          mime-type="text/css">
        </map:read>
      </map:match>
    </map:pipeline>
  </map:pipelines>
</map:sitemap>
```

## Hello World with JavaScript Date

### New File

hello/content/main.xsp

### Modified File

hello/sitemap.xmap

### hello/content/main.jsp

```
<?xml version="1.0"?>
```

```
<xsp:page language="javascript"  
xmlns:xsp="http://apache.org/xsp">
```

```
<xsp:logic>  
  Date now = new Date();  
</xsp:logic>
```

```
<page>  
  <title>Hello World</title>  
  <content>  
    <paragraph>Hi Dad. Today is <xsp:expr>now</xsp:expr>  
  </paragraph>  
  </content>  
</page>  
</xsp:page>
```

## hello/sitemap.xmap

```
<?xml version="1.0"?>
<map:sitemap xmlns:map="http://apache.org/cocoon/sitemap/1.0">
  <map:components>
    <map:generators default="file" />
    <map:transformers default="xslt" />
    <map:readers default="resources" />
    <map:serializers default="html" />
    <map:selectors default="browser" />
    <map:matchers default="wildcard" />
  </map:components>
  <map:pipelines>
    <map:pipeline>
      <map:match pattern="index.jsp">
        <map:generate type="serverpages"
          src="content/mainScript.xsp" />
        <map:transform type="xslt" src="style/main.xsl" />
        <map:serialize type="html" />
      </map:match>
      <map:match pattern="*.css">
        <map:read type="resource" src="resources/{1}.css"
          mime-type="text/css">
        </map:read>
      </map:match>
    </map:pipeline>
  </map:pipelines>
</map:sitemap>
```

## Hello World & Java Date in Separate Files

### Files Changed

hello/content/main.xsp

hello/style/main.xsl

CATALINA\_HOME/webapps/cocoon/WEB-INF/cocoon.xconf

### hello/content/main.xsp

```
<?xml version="1.0"?>
```

```
<xsp:page language="java"  
  xmlns:xsp="http://apache.org/xsp"  
  xmlns:cs683="http://www.eli.sdsu.edu/cs683">
```

```
<page>  
  <title>Hello World</title>  
  <content>  
    <paragraph>Hi Mom. I today is <cs683:dateTime />  
  </paragraph>  
  </content>  
</page>  
</xsp:page>
```

**hello/style/main.xsl**

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xsp="http://apache.org/xsp"
  xmlns:cs683="http://www.eli.sdsu.edu/cs683">

  <xsl:template match="xsp:page">
    <xsp:page>
      <xsl:apply-templates select="@*" />

      <xsp:structure>
        <xsp:include>java.util.Date</xsp:include>
      </xsp:structure>
      <xsp:logic>Date now = new Date();</xsp:logic>

    <html>
      <head>
        <title>
          <xsl:value-of select="title" />
        </title>
        <link rel="Stylesheet" type="text/css"
          media="screen" href="cs683.css" />
      </head>
      <body>
        <xsl:apply-templates />
      </body>
    </html>
  </xsp:page>
</xsl:template>
```

Continued on Next Page

## hello/style/main.xsl

```
<xsl:template match="cs683:dateTime">
  <xsp:expr>now</xsp:expr>
</xsl:template>

<xsl:template match="@*|node()">
  <xsl:copy>
    <xsl:apply-templates select="@*|node()" />
  </xsl:copy>
</xsl:template>

<xsl:template match="title">
  <h2>
    <xsl:apply-templates />
  </h2>
</xsl:template>

<xsl:template match="paragraph">
  <p>
    <xsl:apply-templates />
  </p>
</xsl:template>
</xsl:stylesheet>
```

## hello/sitemap.xmap

```
<?xml version="1.0"?>
<map:sitemap xmlns:map="http://apache.org/cocoon/sitemap/1.0">
  <map:components>
    <map:generators default="file" />
    <map:transformers default="xslt" />
    <map:readers default="resources" />
    <map:serializers default="html" />
    <map:selectors default="browser" />
    <map:matchers default="wildcard" />
  </map:components>
  <map:pipelines>
    </map:match>
    <map:match pattern="index.xsp">
      <map:generate type="serverpages" src="content/main.xsp"
/>
      <map:transform type="xslt" src="style/main.xsl" />
      <map:serialize type="html" />
    </map:match>
    <map:match pattern="*.css">
      <map:read type="resource" src="resources/{1}.css"
        mime-type="text/css">
      </map:read>
    </map:match>
  </map:pipeline>
</map:pipelines>
</map:sitemap>
```

Not changed, but the example is complex so I included it

**CATALINA\_HOME/webapps/cocoon/WEB-INF/cocoon.xconf**

```
<builtin-logicsheet>  
  <parameter name="prefix" value="cs683"/>  
  <parameter name="uri" value="http://www.eli.sdsu.edu/cs683"/>  
  <parameter name="href"  
value="file:/Library/Tomcat/webapps/cocoon/hello/style/javaMain.x  
sl"/>  
</builtin-logicsheet>
```

The above is added in cocoon.xconf file inside the following tag

```
<target-language name="java">  
  
</target-language>
```