IBM Software Group

Webcontainer Part 1: Concepts and Administration

Gabriel Marquez (glmarque@us.ibm.com)

Thanh Giang (tgiang@us.ibm.com)

WebSphere Application Server level 2 support

Date: September 9th, 2014

WebSphere® Support Technical Exchange

ON DEMAND BUSINESS™

Agenda

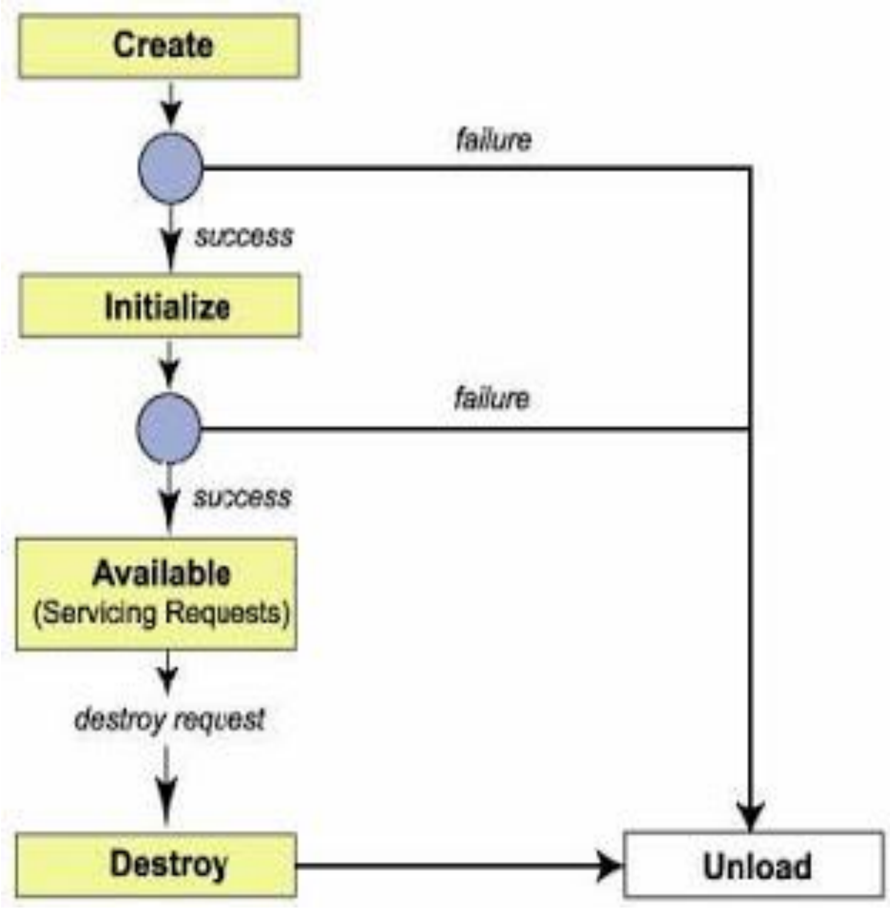
- Java™ Servlet Specification
- Webcontainer settings
- Webcontainer attributes
- Webcontainer custom properties
- Virtual host configuration
- Java EE (J2EE™) application structure and Application XML configuration files

Servlet

- Act as a middle layer between requests from HTTP client (browser) and resources (databases or applications)

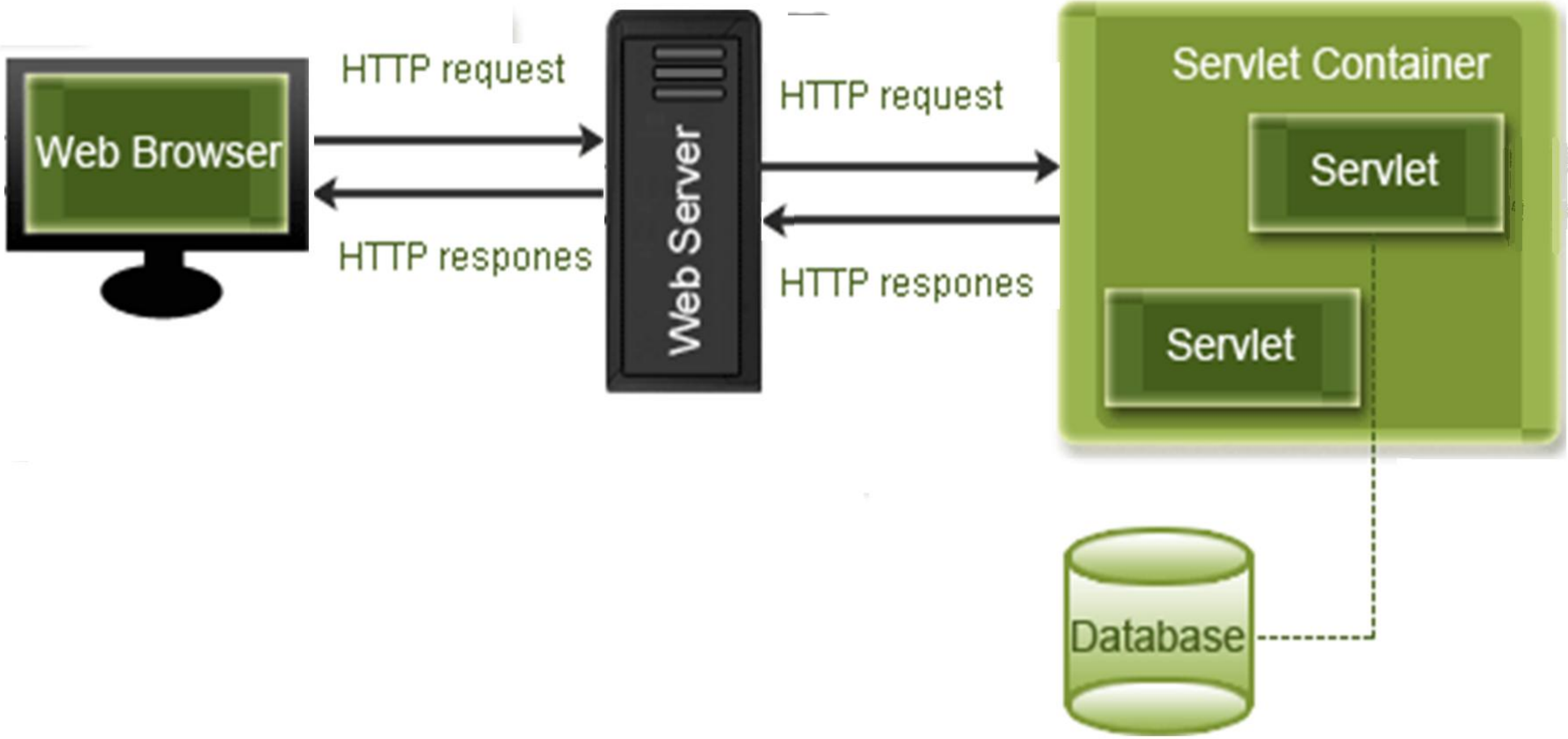
- May perform any of the following tasks:
  - Read
    - Sent data
    - Request data
  - Send
    - Explicit data
    - Implicit response data
  - Generate the results

Servlet Life Cycle

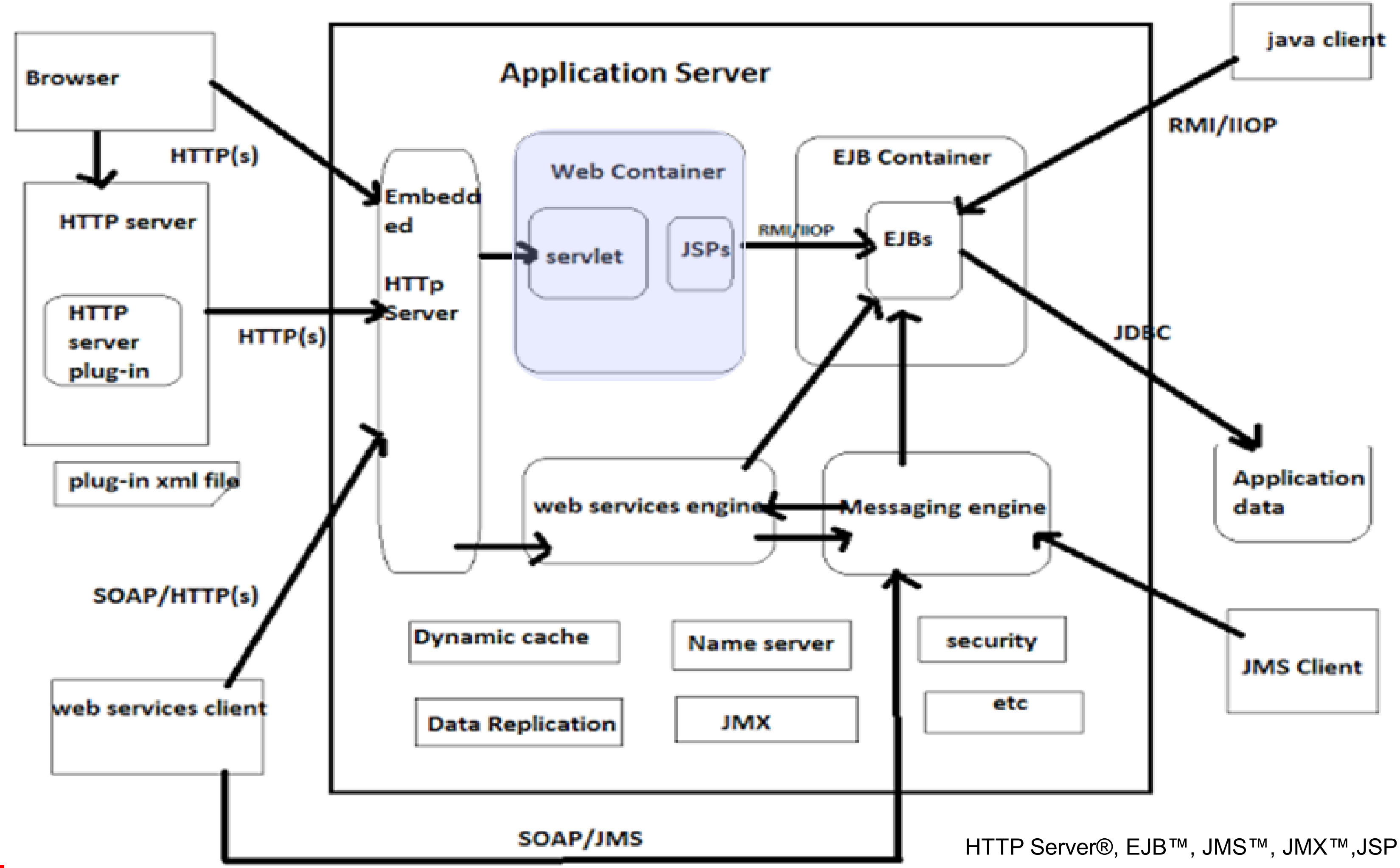


- init()
  - Instantiate / initialize servlet
- service()
  - Execute the servlet
- destroy()
  - Close the resources and remove the servlet from memory

Typical servlet scenario



Webcontainer in Big picture



HTTP Server®, EJB™, JMS™, JMX™,JSP™

Servlet request - Request Path

- requestURI = contextPath + servletPath + pathInfo + query string
  - Context Path – prefix associated with the servlet context
  - Servlet path – mapping to the servlet location
  - Path info – extra path info in URL if any
  - Query string – request parameter name and value pairs

- Example:
  - http://server.com/MyApp/MyServlet/xyz?param1=value1&param2=value2
  - getContextPath - /MyApp
  - getServletPath - /MyServlet
  - getPathInfo - /xyz (many time it's null)
  - getQueryString - param1=value1&param2=value2

Servlet request - Request Life Cycle

- Valid only within the scope of a servlet’s service method or a filter’s doFilter method
- Common to recycle request objects
- References to request objects outside service or doFilter will have indeterminate results e.g.

```
[9/18/13 9:03:31:249 CEST] 000000c5 srt 1 com.ibm.ws.webcontainer.srt.SRTServletRequest getAttribute this-
>com.ibm.ws.webcontainer.srt.SRTServletRequest@8906084a: name --> com.ibm.wps.model.ModelUtil
[9/18/13 9:03:31:249 CEST] 000000c5 srt 1 com.ibm.ws.webcontainer.srt.SRTServletRequest getAttribute this-
>com.ibm.ws.webcontainer.srt.SRTServletRequest@ab65b258: name --> com.ibm.wps.model.impl.LocatorMapperMap
[9/18/13 9:03:31:249 CEST] 000000c5 servlet E com.ibm.ws.webcontainer.servlet.ServletWrapper service SRVE0068E: An
exception was thrown by one of the service methods of the servlet [CustomMenuAction] in application [rd-theme-web-ear].
Exception created : [java.lang.NullPointerException
    at com.ibm.ws.webcontainer.srt.SRTServletRequest$SRTServletRequestHelper.access$200(SRTServletRequest.java:3141)
    at com.ibm.ws.webcontainer.srt.SRTServletRequest.getAttribute(SRTServletRequest.java:366)
    at com.ibm.wps.model.impl.LocatorMapper.getLocator(LocatorMapper.java:179)
```

Servlet response

- Object encapsulating information to be returned from the server to the client

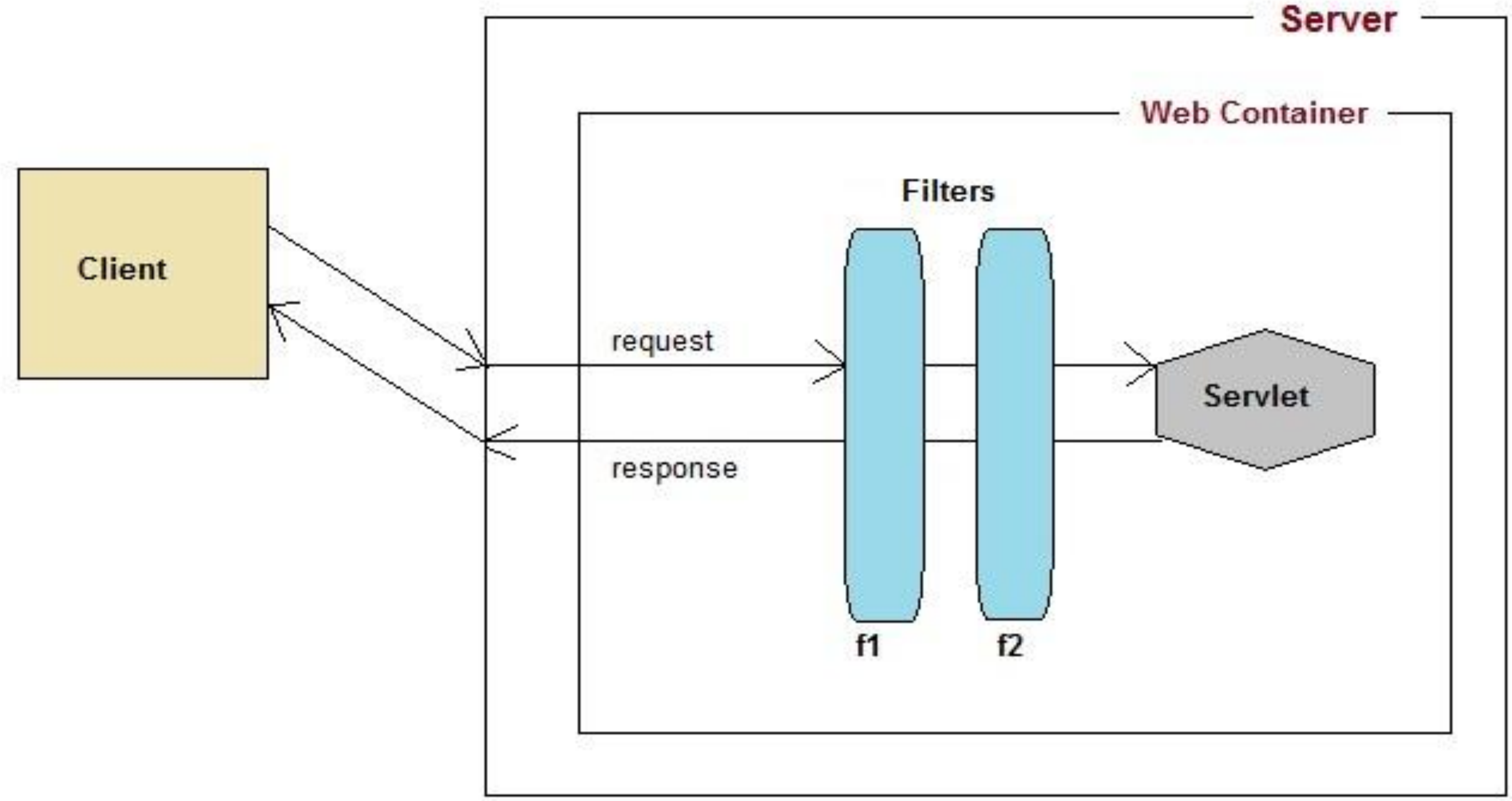
- HTTP protocol - HTTP headers or both headers and a message body of the request
  - Headers
  - Message Body
  - Status Code

- Life Cycle

Servlet response - Closure

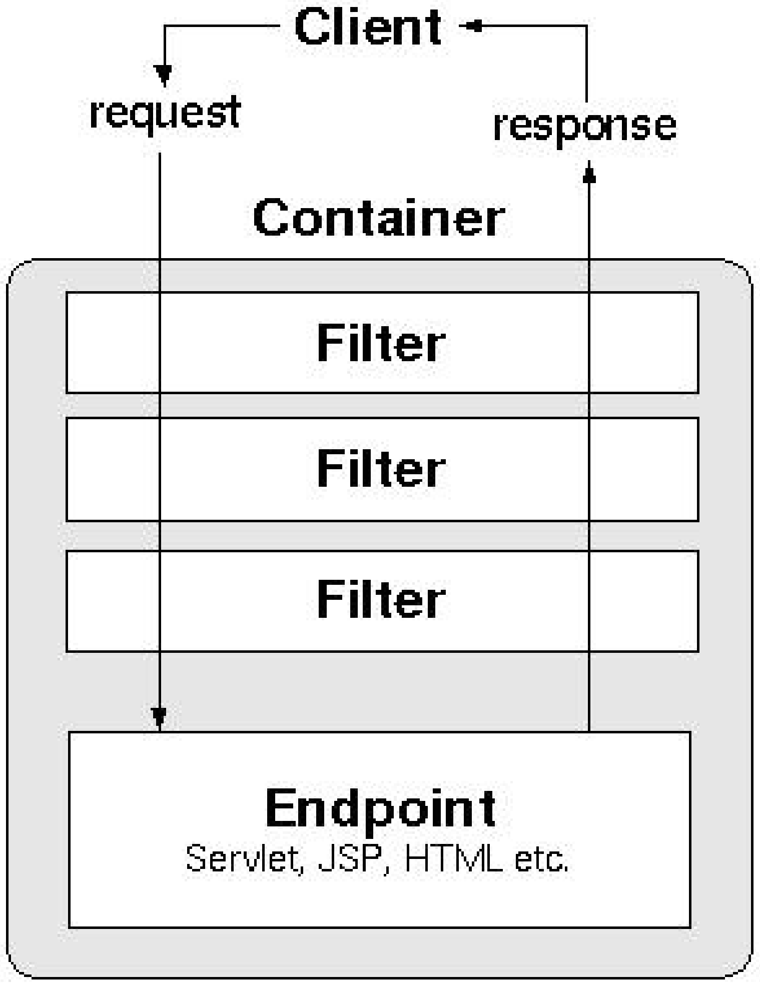
- Container immediately flushes all remaining content in the response buffer to the client
  - Termination of servlet service method
  - Amount of content (setContentLength) has been written to the response
  - The sendError method is called
  - The sendRedirect method is called

Filtering


Server
Web Container
Filters
Client
request
response
Servlet
f1
f2

Filter chain

- Sequence of filters with each filter doing some processing and then passing on to the next in sequence
- Chain member
- Filter chain in the order of their definition in deployment descriptor
- The last element of the chain is the target resource/servlet



Security - Declarative Security

- Application’s security structure stored in a form external to the application
- Deployment descriptor security-constraint tag defines security constrains for web application
- Servlet container uses the security policy representation to enforce authentication and authorization.
- Applies to static content and to servlets and filters

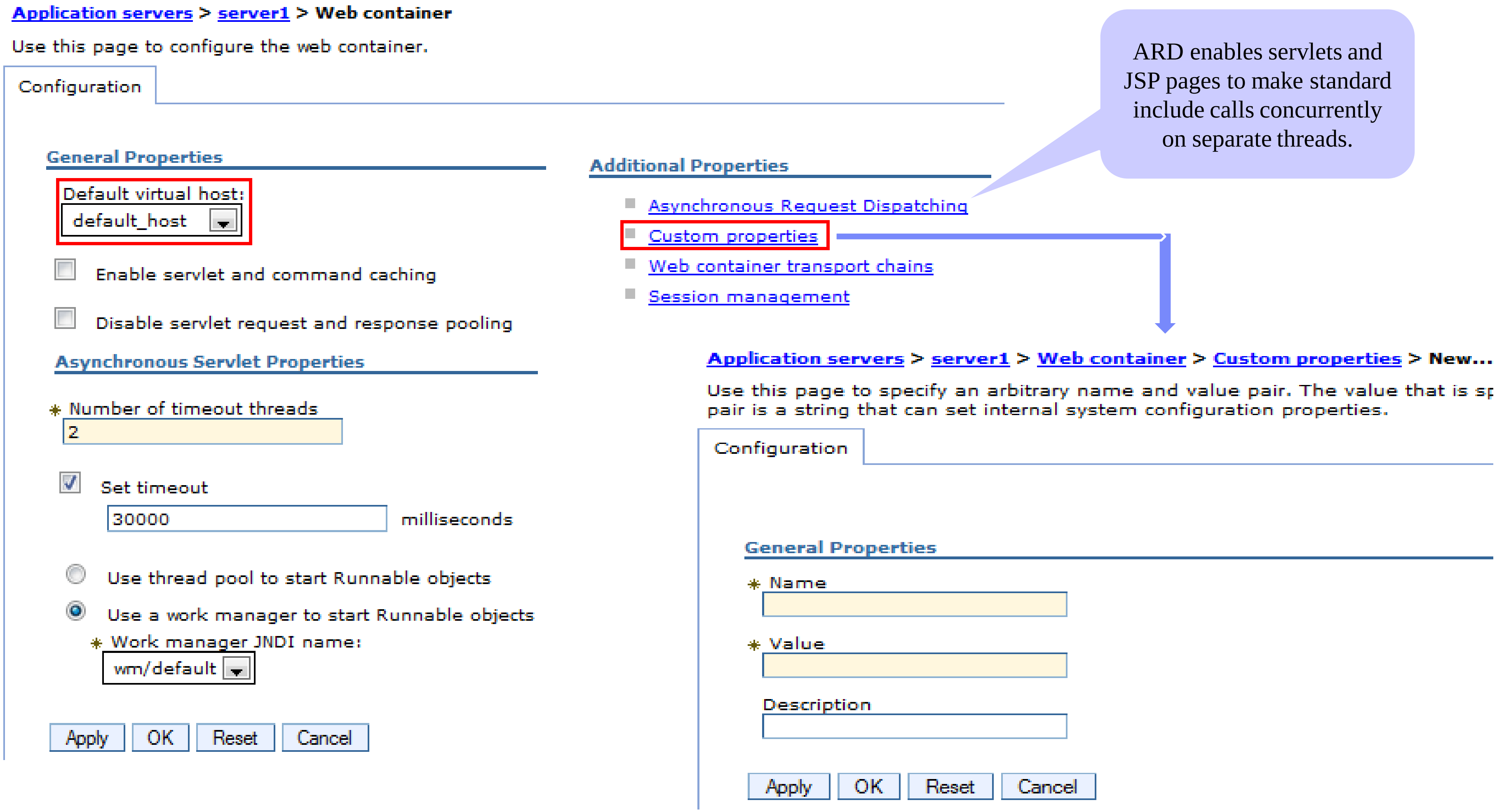
Security - Roles

- Logical grouping of users
- Container enforces security on incoming request
    - Role is mapped to a user group
    - Role is mapped to a principal user

```
<security-role-ref>
    <role-name>FOO</role-name>
    <role-link>manager</role-link>
</security-role-ref>
```

Webcontainer Configuration and Administration

Webcontainer settings



Application servers > server1 > Web container

Use this page to configure the web container.

Configuration

General Properties

Default virtual host:
default_host

□ Enable servlet and command caching

□ Disable servlet request and response pooling

Asynchronous Servlet Properties

* Number of timeout threads
2

☑ Set timeout
30000 milliseconds

○ Use thread pool to start Runnable objects

● Use a work manager to start Runnable objects

* Work manager JNDI name:
wm/default

Additional Properties

- Asynchronous Request Dispatching
- Custom properties
- Web container transport chains
- Session management

ARD enables servlets and JSP pages to make standard include calls concurrently on separate threads.

Application servers > server1 > Web container > Custom properties > New...

Use this page to specify an arbitrary name and value pair. The value that is sp
pair is a string that can set internal system configuration properties.

Configuration

General Properties

* Name

* Value

Description

Webcontainer attributes

The Webcontainer attributes are configured at the application level in the the **ibm-web-ext.xmi** or **ibm-web-ext.xml** file. They are only used by specific application.

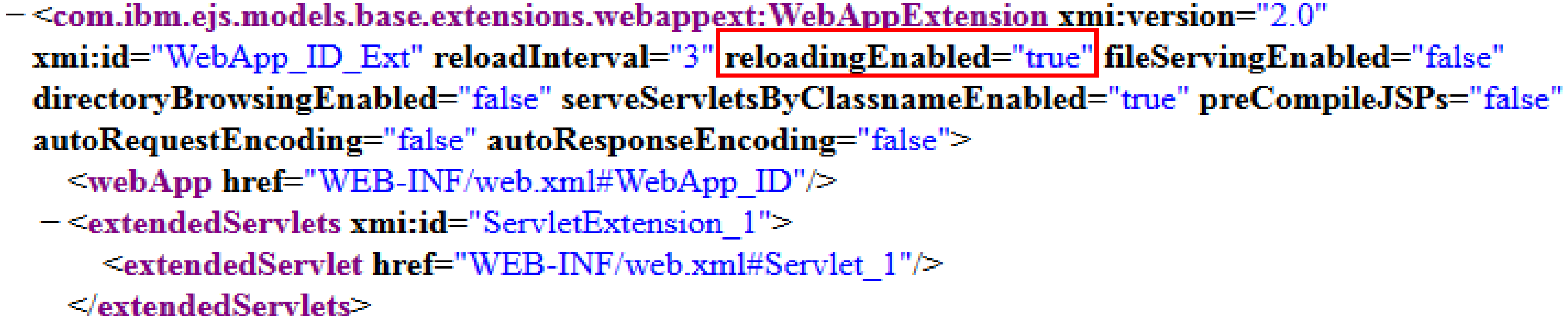
Name	Default Value	Description
fileServingEnabled	true	Allow web applications to serve static file types, such as HTML. If it is set to false, the webserver serves the requests for static files instead.
directoryBrowsingEnabled	false	When this custom property set to true, the application server will display the content of .war file (Excluding the WEB-INF folder) and let you browse through it.

Webcontainer attributes (cont)

Name	Default Value	Description
serveServletsByClassnameEnabled	false	When setting this custom property to true, it allows a servlet to be served via a URI by it class package and class name. This introduces a possible security exposure because the URI pattern which reveals both the fact that your Internet application is a Java Servlet and its Java Class name is basically making public more information than is required To enable the serving of servlets by class name, the new custom property, **com.ibm.ws.webcontainer.disallowserveservletsbyclassname,** must be set to false(default) and **serveServletsByClassnameEnabled** must be enabled for the application which provides the classes to be served. The below technote provides more details about this security vulnerability: http://www.ibm.com/support/docview.wss?uid=swg21288860

Webcontainer attributes (cont)

Name	Default Value	Description
extendedDocumentRoot	none	Use the extended document root facility when applications require access to files outside of the application web application archive (WAR) directory. This facility enables you to configure an application with one or more directory paths from which you can serve static files and JSP files. For example, if several applications require access to a set of common files, you can place the common files in a directory to which you can link each application as an extended document root directory.
reloadingEnabled	true	Specifies whether to enable class reloading when application files (servlet and JSP) are updated.



```
– <com.ibm.ejs.models.base.extensions.webappext:WebAppExtension xmi:version="2.0"
  xmi:id="WebApp_ID_Ext" reloadInterval="3" reloadingEnabled="true" fileServingEnabled="false"
  directoryBrowsingEnabled="false" serveServletsByClassnameEnabled="true" preCompileJSPs="false"
  autoRequestEncoding="false" autoResponseEncoding="false">
    <webApp href="WEB-INF/web.xml#WebApp_ID"/>
  – <extendedServlets xmi:id="ServletExtension_1">
      <extendedServlet href="WEB-INF/web.xml#Servlet_1"/>
    </extendedServlets>
```

Webcontainer custom properties

Unlike the Webcontainer attributes, the Webcontainer custom properties can be configured in the administrative console at the JVM™ level. They affect all applications running in the server.

Name	Default Value	Description
com.ibm.ws.webcontainer.disallowAllFileServing	false	Enabling this custom property to disable file serving on all applications on a specific application server. This prevents the application server from serving static files.
com.ibm.ws.webcontainer.disallowserveservletsbyclassname	false	Disallow to access servlets directly which results in a possible security exposure. To enable the serving of servlets by class name the new custom property ***com.ibm.ws.webcontainer.disallowserveservletsbyclassname*** must be set to false (default) and ***serveServletsByClassnameEnabled*** must be enabled for the application which provides the classes to be served.
com.ibm.ws.webcontainer.donotservebyclassname	none	Specifies a list of classes that cannot be served by the class name.

Webcontainer custom properties (cont)

Name	Default Value	Description
com.ibm.ws.webcontainer.extractHostHeaderPort	false	The getServerPort method relies on the getVirtualPort method of the channel, which returns a port number in the following order: Port number from the request URL Port number from the request host header This order is compliant with HTTP/1.1 RFC but not with the Java Servlet Specification Version 2.4 API, which requires the port number from the host header to be returned first, if any, or the request URL. The correct returned URL for the above example is: http://ProxyServer:8888. The Webcontainer was modified to return a port number from the host header, if any, or the URL port that accepted the client connection. You must set the trusthostheaderport and the com.ibm.ws.webcontainer.extractHostHeaderPort custom property to true to return the port number from the request host header first.
trusthostheaderport	false	

Webcontainer custom properties (cont)

Name	Default Value	Description
com.ibm.ws.webcontainer.channelwritetype	async	By default, the Webcontainer uses asynchronous writes to write response data in chunks up to the response buffer size. If the response is more than the response buffer size, the Webcontainer continues to buffer response data into memory while waiting for an asynchronous write of a response data chunk to complete. This process can result in part of a large response held in memory, which can lead to high memory usage and potentially an out of memory error. To prevent out of memory issue, set this property to **sync** so synchronous writing is used. With synchronous writing, response data are written synchronously in chunks of up to the value of responsebuffersize and no response data are buffered into memory while waiting for a synchronous write of a response data chunk to complete.

Webcontainer custom properties (cont)

Name	Default Value	Description
com.ibm.ws.webcontainer.SkipMetaInfResourcesProcessing	false	The Webcontainer searches for static files and JavaServer Pages (JSP) files in different locations, depending on application configuration. A web fragment comprises a JAR file in an application WEB-INF/lib directory. The JAR might include static resources in a META-INF/resources directory that are defined within the JAR file. Searching for static resources in the web fragment JAR files is required by the Servlet Specification 3.0. If the application is not relying on the Servlet 3.0 searching behavior, you can disable the META-INF resource searching by setting this custom property to true. Enable this custom property would help to improve performance.

Webcontainer custom properties (cont)

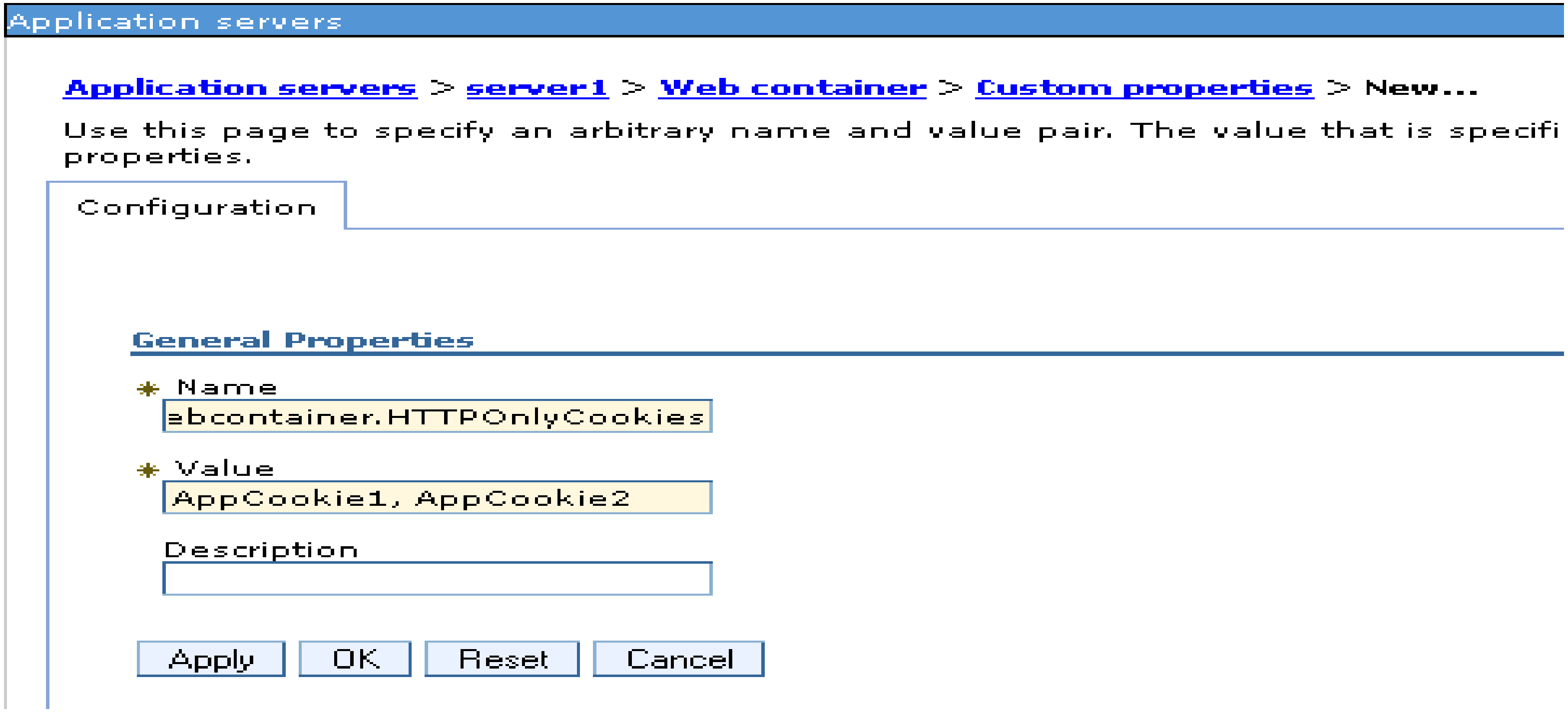
Name	Default Value	Description
com.ibm.ws.webcontainer.HTTPOnlyCookies ye	none Property value: a list of case-insensitive application cookie names which are separated by a comma.	Provides a level of defense against a client-side script accessing a protected cookie and acquiring its content. When you use this custom property, you can prevent Java scripts that run in a browser from accessing all cookies or a particular list of cookies of your choosing. The HTTPOnly attribute is added to each cookie specified in this custom property and enables protection from client-side script access. To disable, **Application servers** > *server_name* > **Webcontainer** > **Session management** > **Cookies** **-** Uncheck the checkbox "***Set session cookies to HTTPOnly to help prevent cross-site scripting attacks***"

More Webcontainer customer properties can be found at:

http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-nd-dist&topic=rweb_custom_props

Configuring Webcontainer custom properties in the administrative console

The below example shows how to configure the com.ibm.ws.webcontainer.HTTPOnlyCookies custom property for the application cookies called “AppCookie1, AppCookie2”.



Application servers

Application servers > server1 > Web container > Custom properties > New...

Use this page to specify an arbitrary name and value pair. The value that is specifi properties.

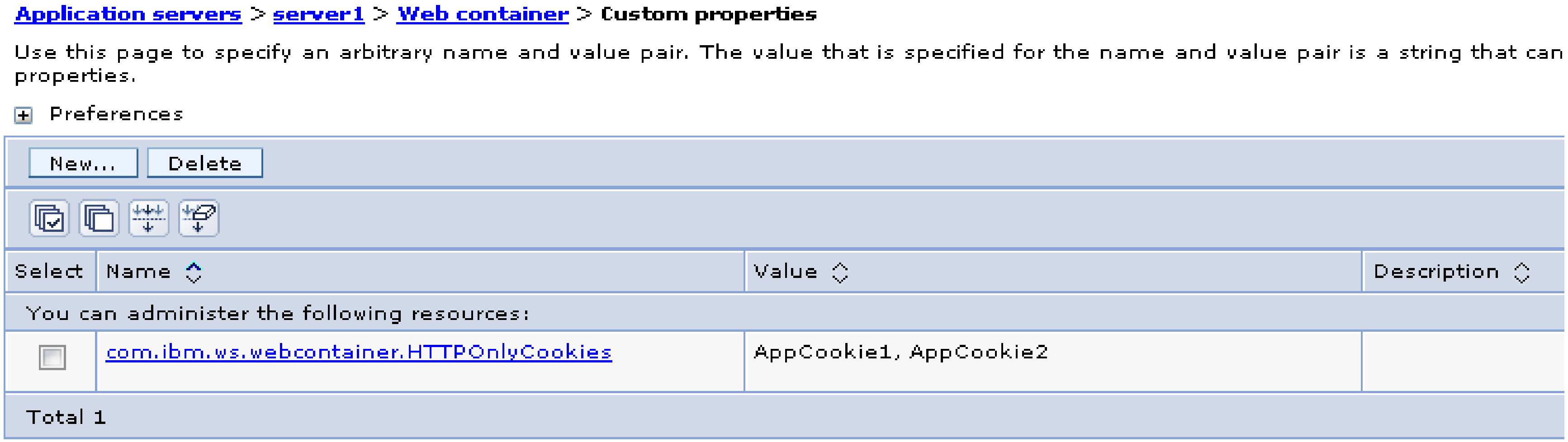
Configuration

General Properties

* Name: ebcontainer.HTTPOnlyCookies

* Value: AppCookie1, AppCookie2

Description:



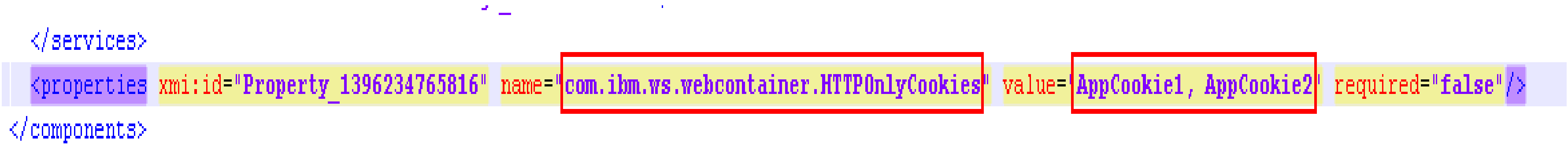
Application servers > server1 > Web container > Custom properties

Use this page to specify an arbitrary name and value pair. The value that is specified for the name and value pair is a string that can
properties.

⊞ Preferences

Select	Name	Value	Description
You can administer the following resources:			
□	com.ibm.ws.webcontainer.HTTPOnlyCookies	AppCookie1, AppCookie2	
Total 1			

The configured Webcontainer properties can be found in the **server.xml** file:



```
  </services>
  <properties xmi:id="Property_1396234765816" name="com.ibm.ws.webcontainer.HTTPOnlyCookies" value="AppCookie1, AppCookie2" required="false"/>
</components>
```

Virtual host

- A virtual host is a configuration entity that allows WebSphere Application Server to treat multiple host names, IP addresses, or port numbers as a single logical host. It is associated with a particular application Server.

- By default, an application server profile provides a ***default virtual host*** with some common host aliases such as the IP address, the DNS short host name, and the DNS fully qualified host name.

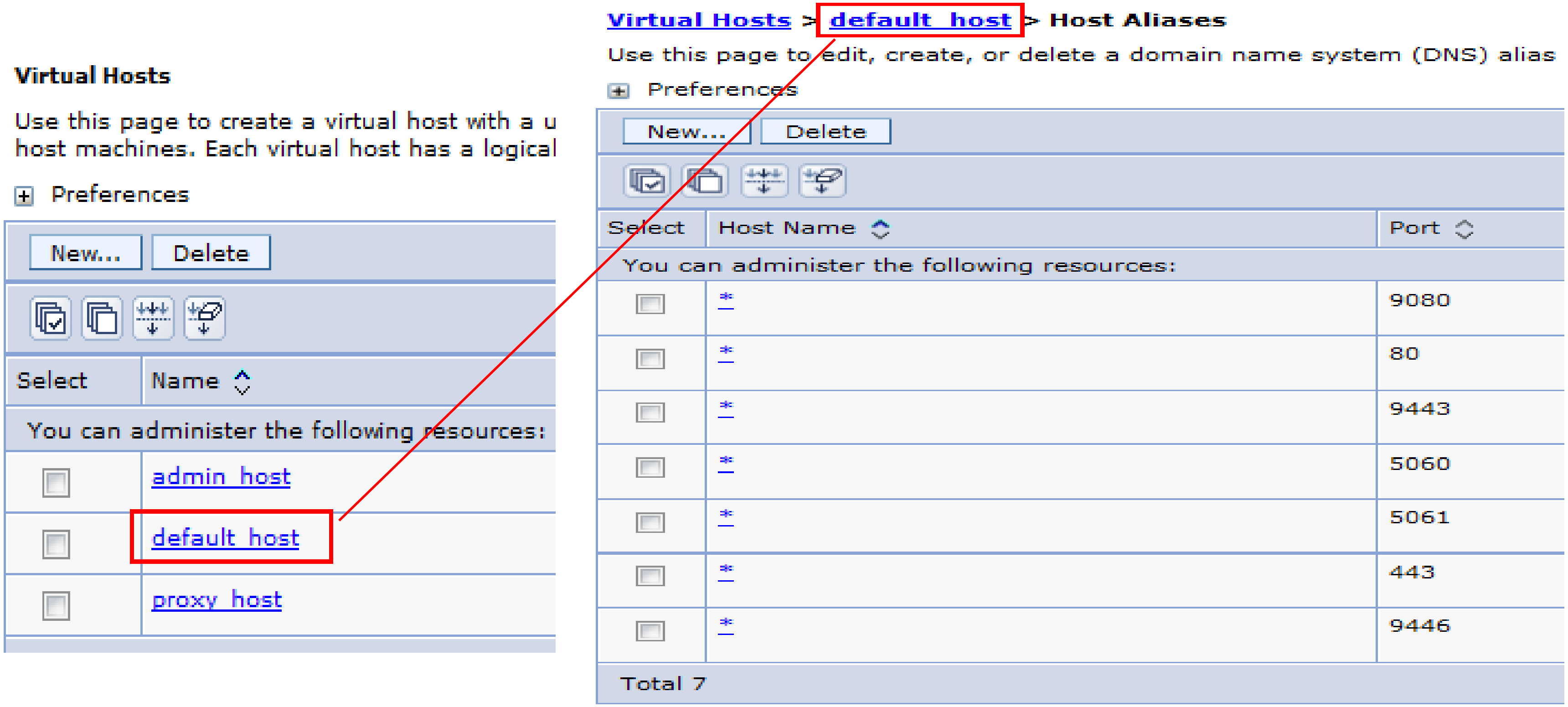
- Each virtual host contains a list of one or more domain name system (DNS) aliases. A DNS alias is the TCP/IP host name and port number used to request the servlet. For example: The alias is myhost:9080 in the request http://myhost:9080/HelloServlet

- The host alias must be unique in all virtual hosts configured in the server. Otherwise, the request might be sent to the wrong application server and the "**Virtual Host or Web Application Not Found: The host *alias* has not been defined**" would be generated.

- You can associate a virtual host to one or more Web modules, but you can associate each Web module with one and only one virtual host.

Virtual host configuration for default_host

The default virtual host, named **default_host**, is automatically configured the first time you start an application server. The below diagram shows the host aliases defined in the virtual host “**default_host**”:



Virtual Hosts

Use this page to create a virtual host with a u
host machines. Each virtual host has a logical

⊞ Preferences

Select	Name
You can administer the following resources:	
□	admin_host
□	default_host
□	proxy_host

Virtual Hosts > default_host > Host Aliases

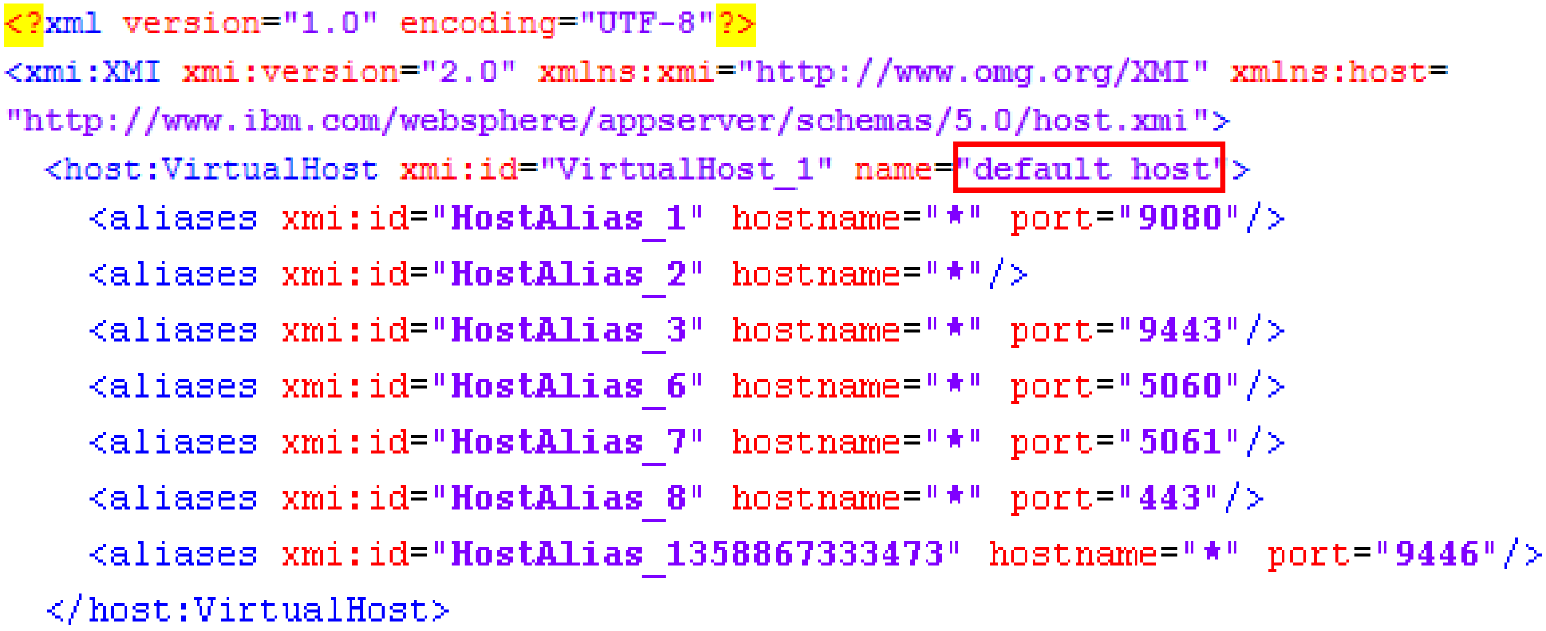
Use this page to edit, create, or delete a domain name system (DNS) alias

⊞ Preferences

Select	Host Name	Port
You can administer the following resources:		
□	*	9080
□	*	80
□	*	9443
□	*	5060
□	*	5061
□	*	443
□	*	9446
Total 7		

Virtual host configuration for default_host in the virtualhost.xml file

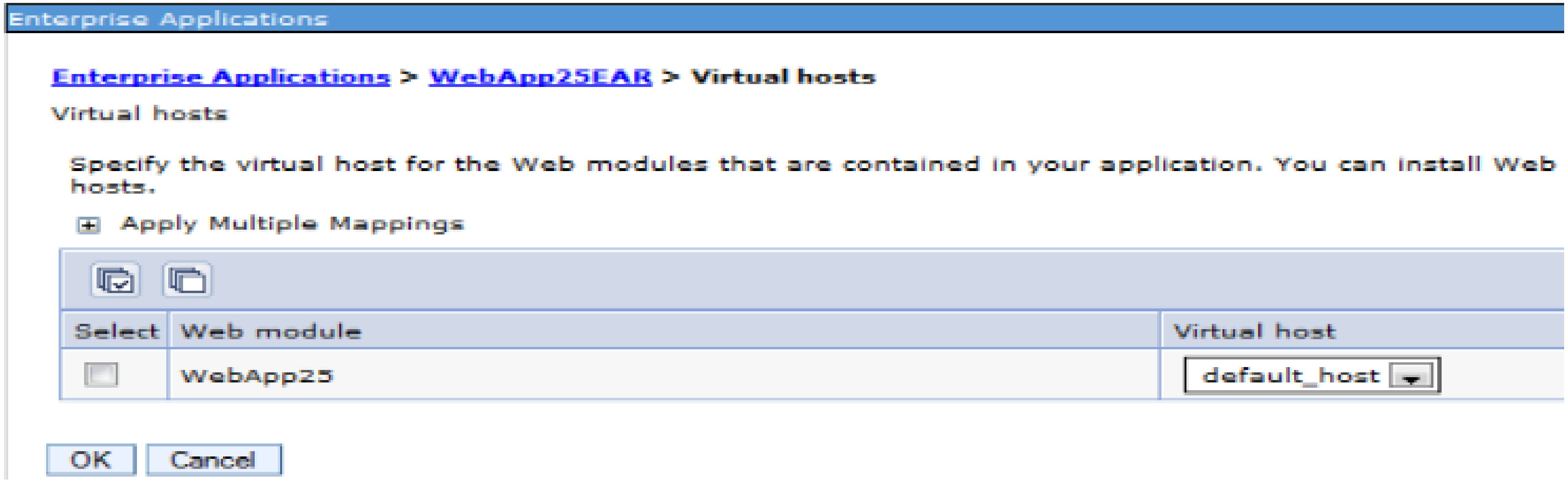
The virtualhost.xml file can be found in *{WAS_ROOT}/profiles/profilename/config/cells/cellname/* directory. You can find what host aliases are defined for specific virtual host in the virtualhost.xml file.



```
<?xml version="1.0" encoding="UTF-8"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:host=
"http://www.ibm.com/websphere/appserver/schemas/5.0/host.xmi">
  <host:VirtualHost xmi:id="VirtualHost_1" name="default_host">
    <aliases xmi:id="HostAlias_1" hostname="*" port="9080"/>
    <aliases xmi:id="HostAlias_2" hostname="*"/>
    <aliases xmi:id="HostAlias_3" hostname="*" port="9443"/>
    <aliases xmi:id="HostAlias_6" hostname="*" port="5060"/>
    <aliases xmi:id="HostAlias_7" hostname="*" port="5061"/>
    <aliases xmi:id="HostAlias_8" hostname="*" port="443"/>
    <aliases xmi:id="HostAlias_1358867333473" hostname="*" port="9446"/>
  </host:VirtualHost>
```

Mapping Virtual hosts for web modules

- Each web module must be mapped to a previously defined virtual host, identified under Virtual host.
- From the administrative console, select **Applications > Application Types > WebSphere enterprise applications > application_name > Virtual hosts** to find out what virtual host the application is mapped to.



Enterprise Applications

Enterprise Applications > WebApp25EAR > Virtual hosts

Virtual hosts

Specify the virtual host for the Web modules that are contained in your application. You can install Web hosts.

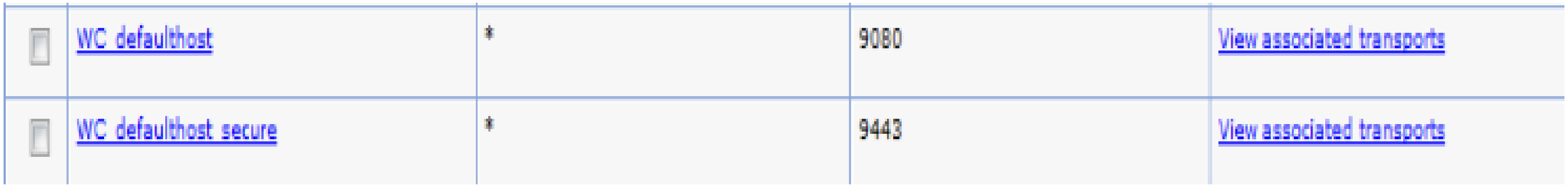
⊞ Apply Multiple Mappings

Select	Web module	Virtual host
□	WebApp25	default_host

Webcontainer communication ports

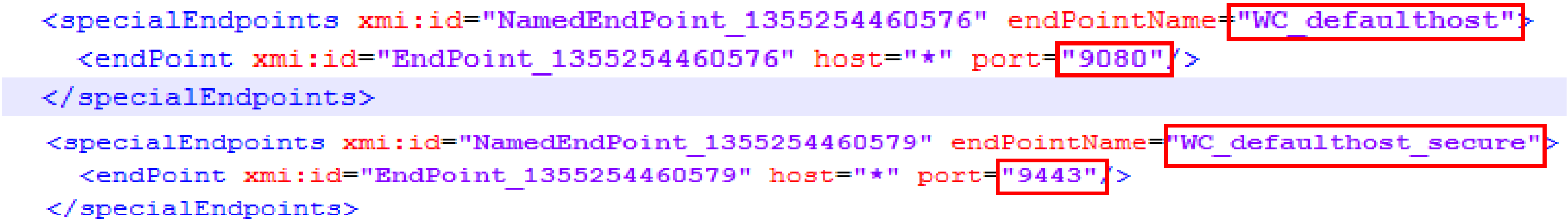
There are two ways to find out the communication ports that the Webcontainer is listening to.

1. From the administrative console, click on **Application Servers > servername > Ports**. The below diagram shows the Webcontainer communication ports is listening to port 9080 (non-secure) and 9443 (secure).



☐	WC_defaulthost	*	9080	View associated transports
☐	WC_defaulthost_secure	*	9443	View associated transports

2. Another way is looking at the serverindex.xml which locates in the *{WAS_ROOT}/profiles/profilename/config/cells/cellname/nodes/nodename/* directory.



```
<specialEndpoints xmi:id="NamedEndPoint_1355254460576" endPointName="WC_defaulthost">
  <endPoint xmi:id="EndPoint_1355254460576" host="*" port="9080"/>
</specialEndpoints>
<specialEndpoints xmi:id="NamedEndPoint_1355254460579" endPointName="WC_defaulthost_secure">
  <endPoint xmi:id="EndPoint_1355254460579" host="*" port="9443"/>
</specialEndpoints>
```

Java EE application

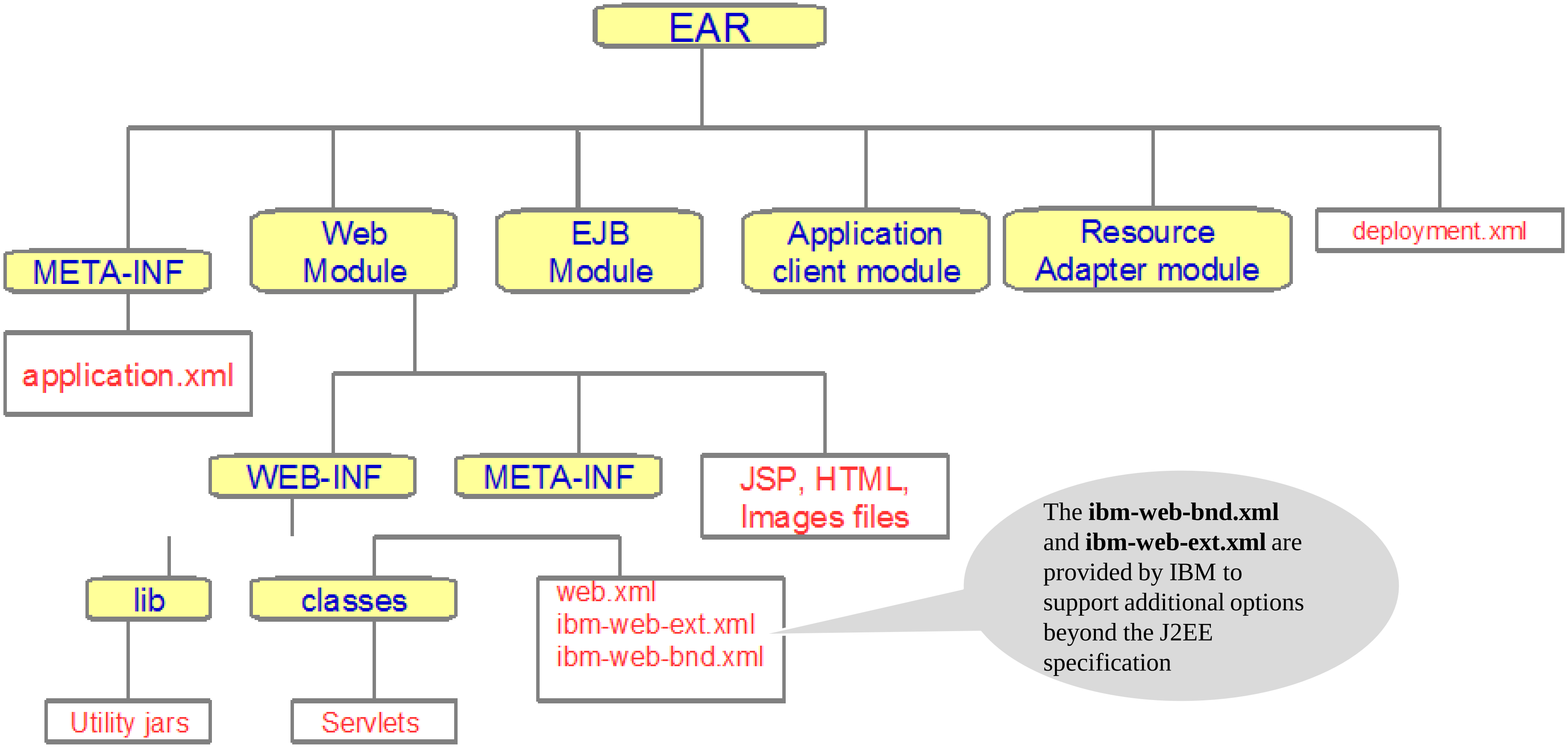
- Java EE application is packaged in an Enterprise Archive (EAR) file.

- An EAR file contains web modules, enterprise bean (EJB) modules, resource adapters, Java EE application client, utility jars, and deployment descriptors.

- The web module packages into a WAR file, and EJBs are packaged into a JAR. Both WAR and EJB jars are packaged into the EAR file.

- The utility jars or third party jars files can be packaged at the root of the EAR file. The jar files can be shared by all the modules in the EAR.

Java EE application structure



The **ibm-web-bnd.xml** and **ibm-web-ext.xml** are provided by IBM to support additional options beyond the J2EE specification

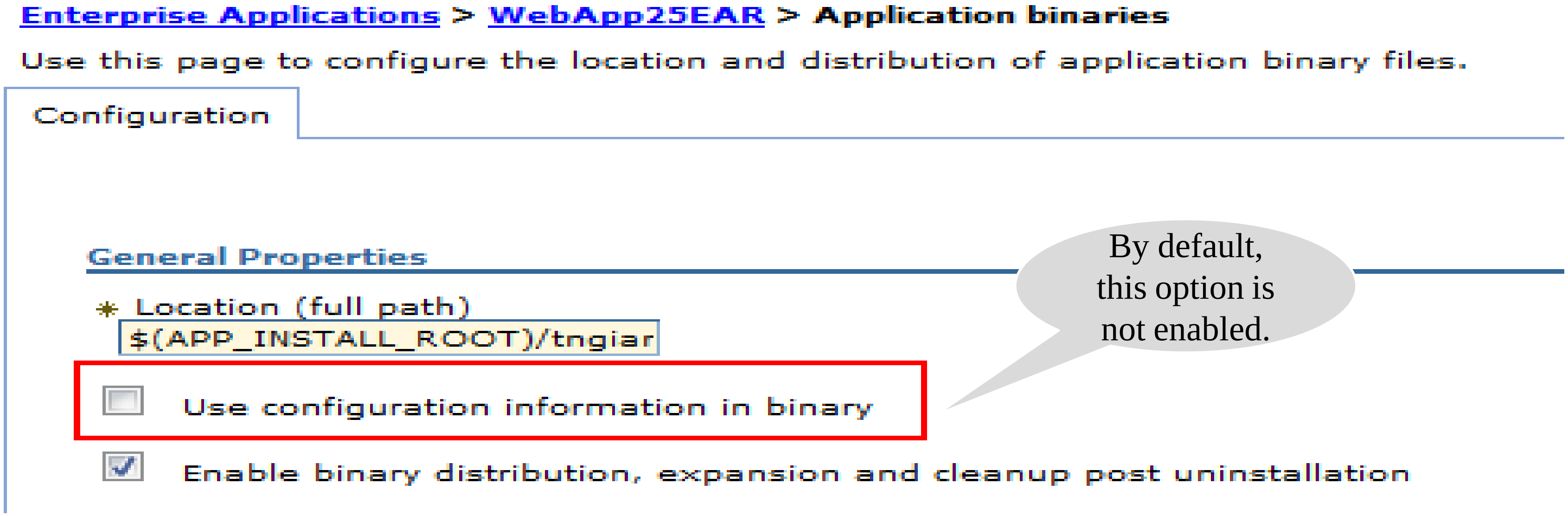
Application XML configuration files

- By default, an EAR file is installed in the *profile_root/installedApps/cell_name/application_name.ear* directory. This directory contains the servlets, JSPs and XML files.

- During runtime, the server reads the XML files in the *profile_root/config/cells/cellname/applications/application_name directories* by default. If you require to modify the XML files after the application is installed you would need to modify them in the ***config*** directory and not in the ***installedApps*** directory.

- If you want the application server uses the binding, extensions, and deployment descriptors in the ***installedApps*** directory enable the "**Use configuration information in binary"** option (see diagram on page 31):

Application XML configuration files (cont)



Enterprise Applications > WebApp25EAR > Application binaries

Use this page to configure the location and distribution of application binary files.

* Location (full path)
$(APP_INSTALL_ROOT)/tngiar

□ Use configuration information in binary

☑ Enable binary distribution, expansion and cleanup post uninstallation

By default, this option is not enabled.

Application XML configuration files (cont)

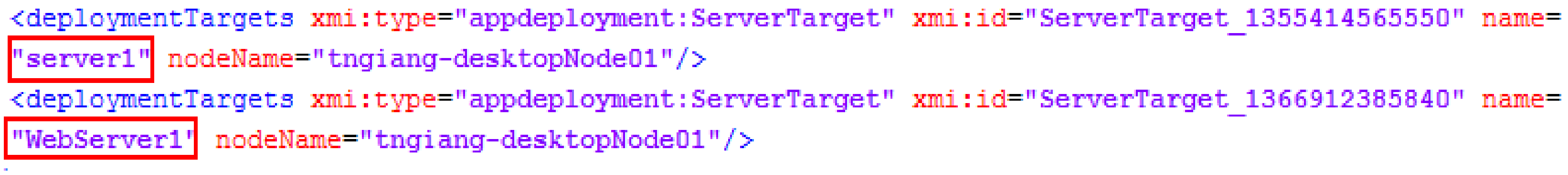
Some of the application XML files can be viewed in the administrative console such as the application.xml and web.xml files:

Name	Description	Accessing a console view
application.xml	Context root of the web application	**Applications** > **Application Types** > **WebSphere enterprise applications** > *application_name* > **View deployment descriptor**
web.xml		**Applications** > **Application Types** > **WebSphere enterprise applications** > *application_name* > **Manage modules** > *module_name* **> View deployment descriptor**

NOTE: Any changes made to the application XML files require the application restart.

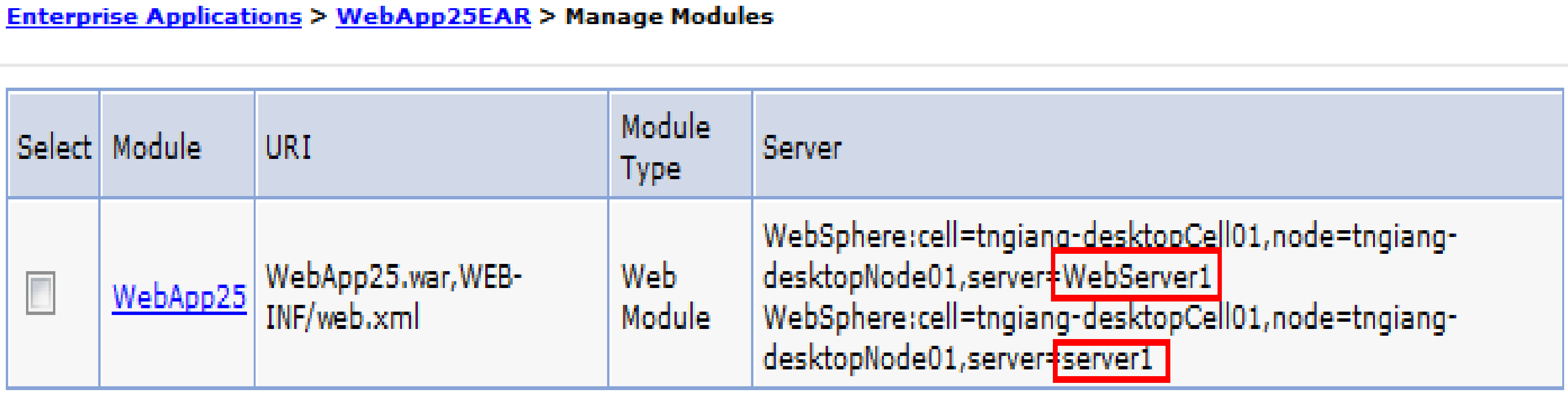
deployment.xml

- Each web module must be mapped to one or more targets (servers).
- To find out what servers the web module is mapped to review the deployment.xml file in the *{WAS_ROOT}/profiles/profilename/config/cells/cellname/ applications/enterpriseAppName/deployments/deployedName* directory.



```
<deploymentTargets xmi:type="appdeployment:ServerTarget" xmi:id="ServerTarget_1355414565550" name=
"server1" nodeName="tngiang-desktopNode01"/>
<deploymentTargets xmi:type="appdeployment:ServerTarget" xmi:id="ServerTarget_1366912385840" name=
"WebServer1" nodeName="tngiang-desktopNode01"/>
```

- You can find the same details from the administrative console below:

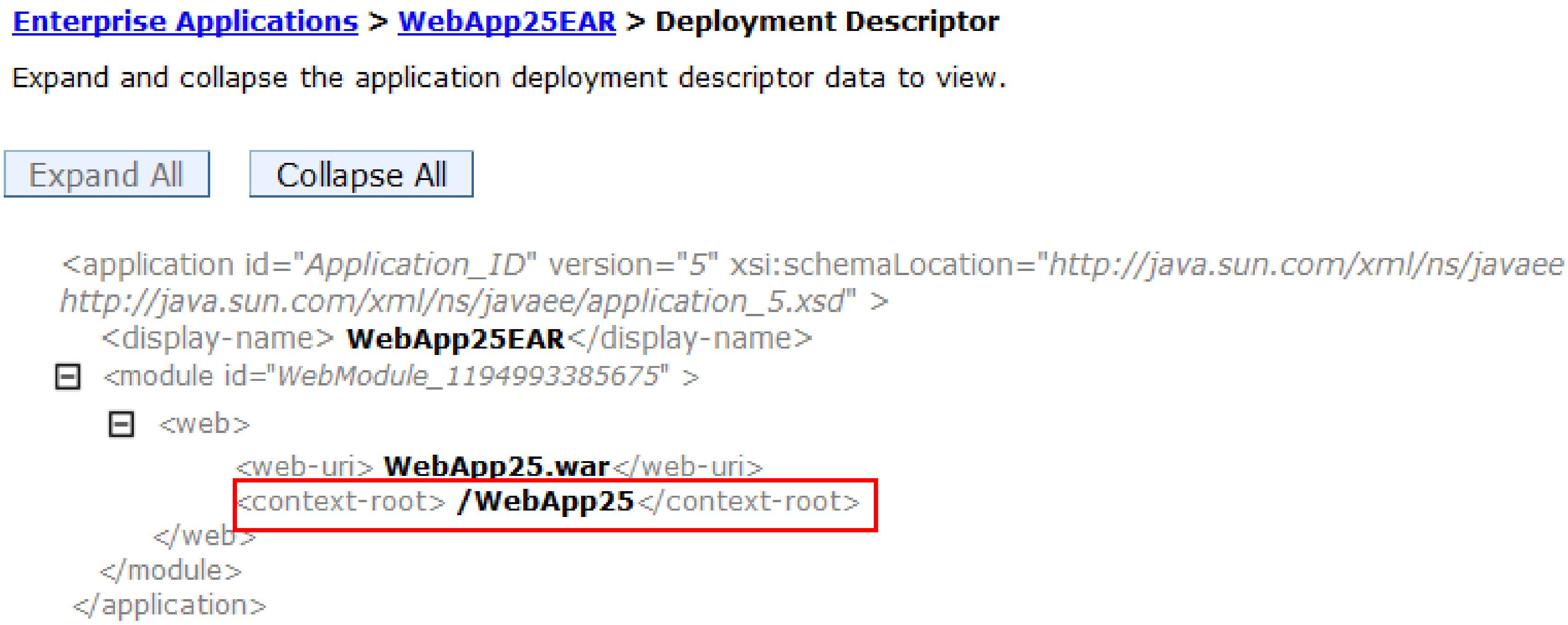


Enterprise Applications > WebApp25EAR > Manage Modules

Select	Module	URI	Module Type	Server
□	WebApp25	WebApp25.war,WEB-INF/web.xml	Web Module	WebSphere:cell=tngiang-desktopCell01,node=tngiang-desktopNode01,server=WebServer1 WebSphere:cell=tngiang-desktopCell01,node=tngiang-desktopNode01,server=server1

application.xml

The context root for the application can be found in the application.xml file in the ***{WAS_ROOT}/profiles/profilename/config/cells/cellname/applications/enterpriseAppName/deployments/deployedName/META-INF*** directory or from the administrative console below:



Enterprise Applications > WebApp25EAR > Deployment Descriptor

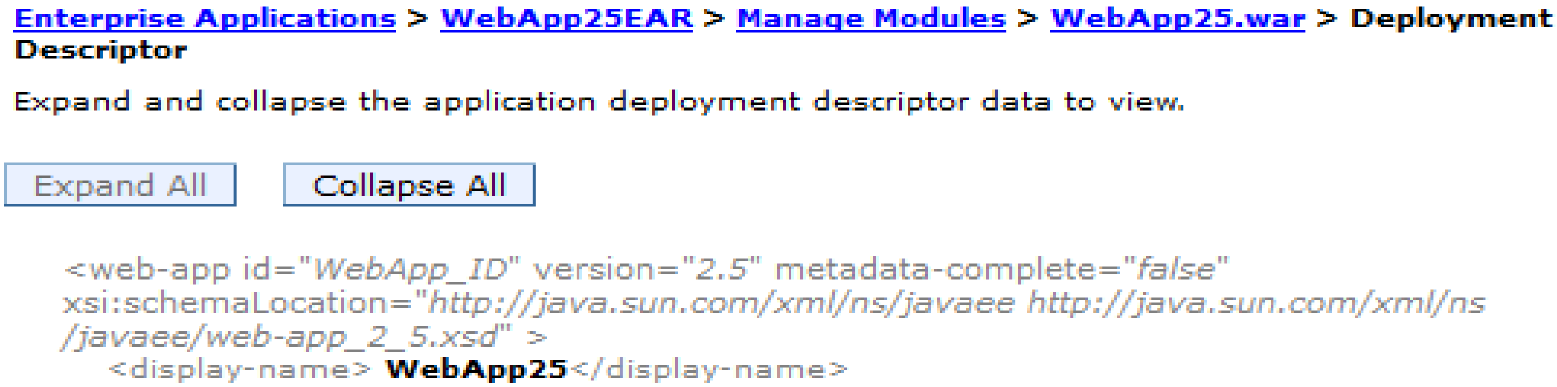
Expand and collapse the application deployment descriptor data to view.

```
<application id="Application_ID" version="5" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/application_5.xsd" >
  <display-name> WebApp25EAR</display-name>
⊟ <module id="WebModule_1194993385675" >
   ⊟ <web>
         <web-uri> WebApp25.war</web-uri>
         <context-root> /WebApp25</context-root>
     </web>
  </module>
</application>
```

web.xml

- The web.xml file is the Web Application Deployment Descriptor of your application. It defines everything about your application that a server needs to know such as servlet mapping, filter mapping, listeners, initialization parameters, container-managed security constraints, resources, welcome pages, and error page.

- The web.xml file can be located in the *{WAS_ROOT}/profiles/profilename/config/cells/cellname/applications/enterpriseAppName/deployments/deployedName/webModuleName/WEB-INF* directory or in the administrative console below:

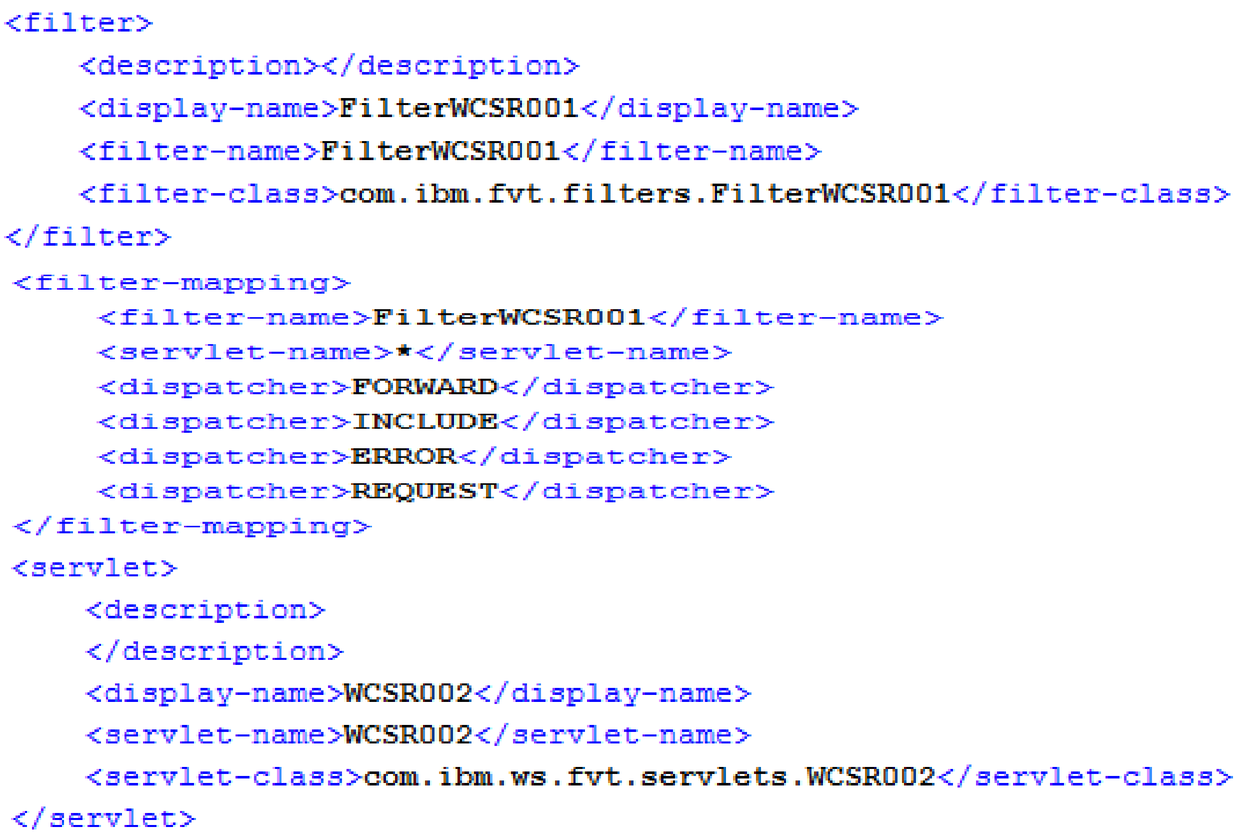


Enterprise Applications > WebApp25EAR > Manage Modules > WebApp25.war > **Deployment Descriptor**

Expand and collapse the application deployment descriptor data to view.

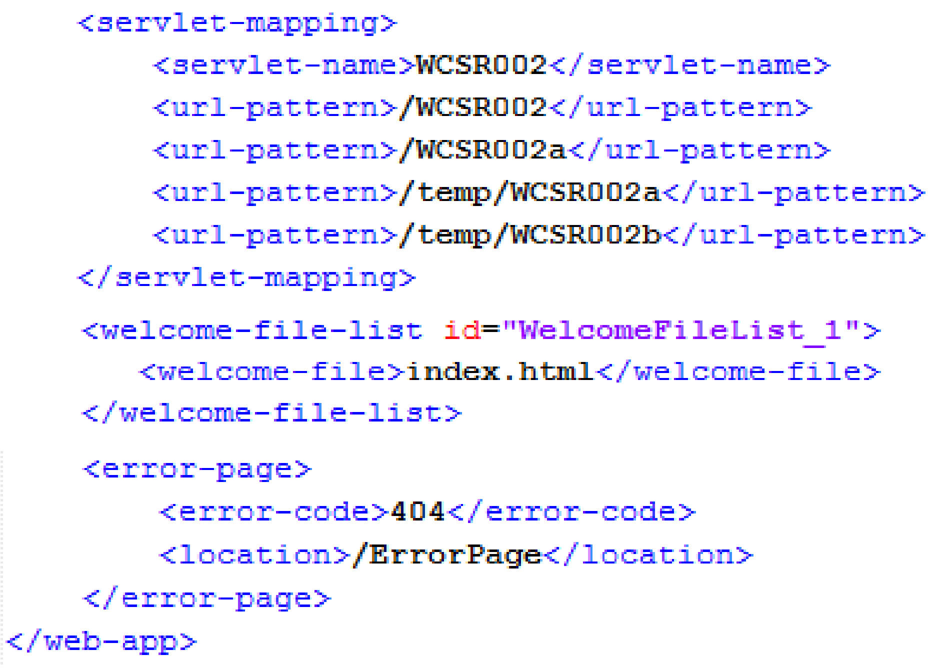
```
<web-app id="WebApp_ID" version="2.5" metadata-complete="false"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns
/javaee/web-app_2_5.xsd" >
   <display-name> WebApp25</display-name>
```

web.xml (cont)



```
<filter>
    <description></description>
    <display-name>FilterWCSR001</display-name>
    <filter-name>FilterWCSR001</filter-name>
    <filter-class>com.ibm.fvt.filters.FilterWCSR001</filter-class>
</filter>
<filter-mapping>
    <filter-name>FilterWCSR001</filter-name>
    <servlet-name>*</servlet-name>
    <dispatcher>FORWARD</dispatcher>
    <dispatcher>INCLUDE</dispatcher>
    <dispatcher>ERROR</dispatcher>
    <dispatcher>REQUEST</dispatcher>
</filter-mapping>
<servlet>
    <description>
    </description>
    <display-name>WCSR002</display-name>
    <servlet-name>WCSR002</servlet-name>
    <servlet-class>com.ibm.ws.fvt.servlets.WCSR002</servlet-class>
</servlet>
```

web.xml (cont)

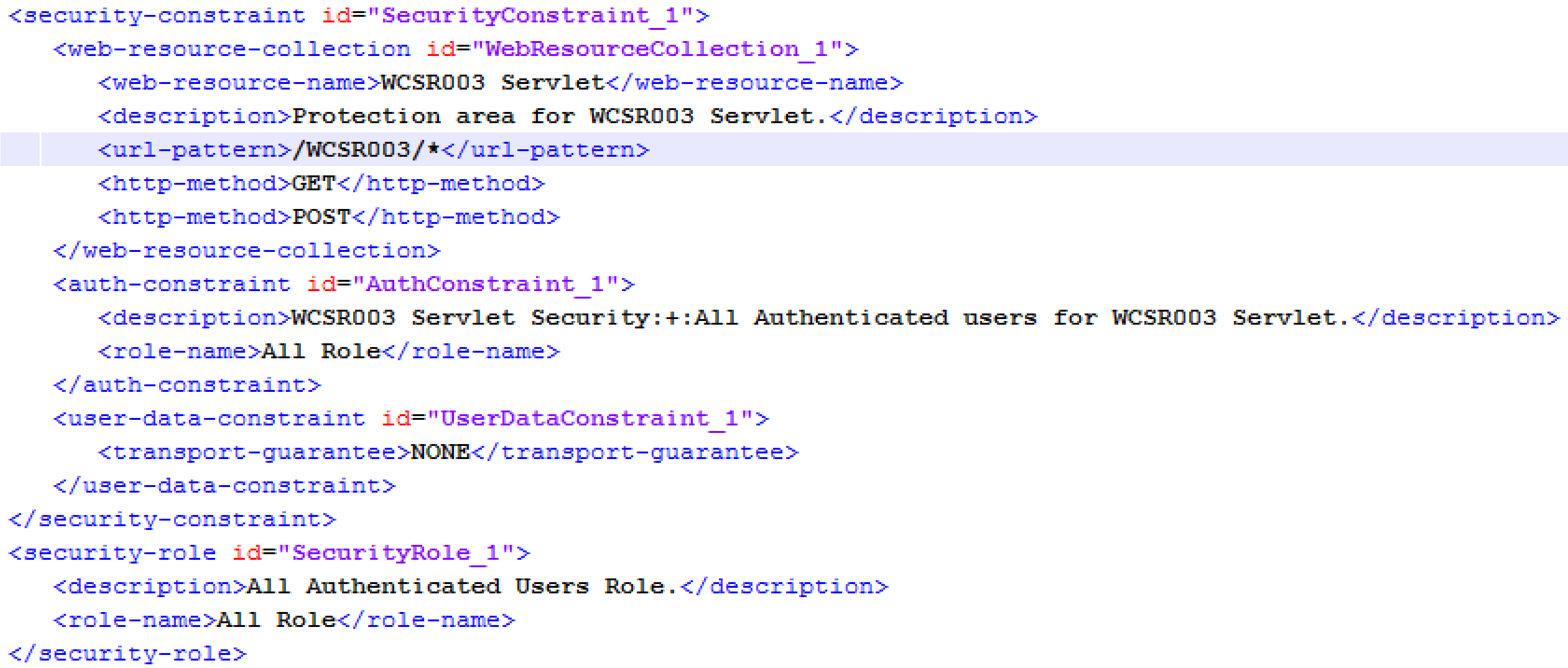


```
    <servlet-mapping>
        <servlet-name>WCSR002</servlet-name>
        <url-pattern>/WCSR002</url-pattern>
        <url-pattern>/WCSR002a</url-pattern>
        <url-pattern>/temp/WCSR002a</url-pattern>
        <url-pattern>/temp/WCSR002b</url-pattern>
    </servlet-mapping>
    <welcome-file-list id="WelcomeFileList_1">
       <welcome-file>index.html</welcome-file>
    </welcome-file-list>
    <error-page>
        <error-code>404</error-code>
        <location>/ErrorPage</location>
    </error-page>
</web-app>
```

web.xml – security constraint

The <security-constraint> element defines a security constraint for URLs that match a pattern. If a user accesses a URL whose path has a security constraint and the user will require to authenticate in order to access the resource.

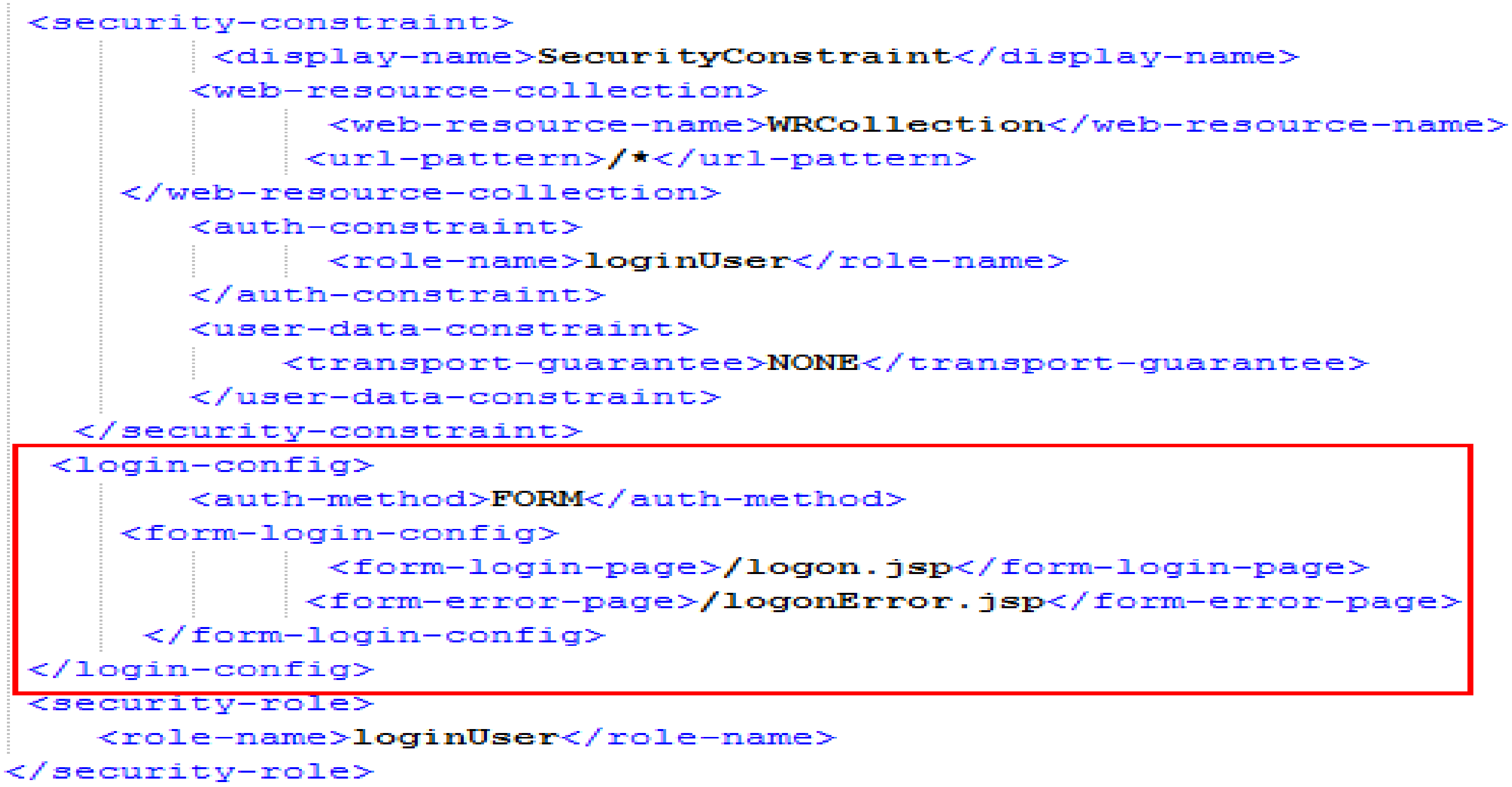
Below is an example of the <security-constraint> element in the web.xml file. All authenticated users can access the URL path /WCSR003/*.



```xml
<security-constraint id="SecurityConstraint_1">
   <web-resource-collection id="WebResourceCollection_1">
      <web-resource-name>WCSR003 Servlet</web-resource-name>
      <description>Protection area for WCSR003 Servlet.</description>
      <url-pattern>/WCSR003/*</url-pattern>
      <http-method>GET</http-method>
      <http-method>POST</http-method>
   </web-resource-collection>
   <auth-constraint id="AuthConstraint_1">
      <description>WCSR003 Servlet Security:+:All Authenticated users for WCSR003 Servlet.</description>
      <role-name>All Role</role-name>
   </auth-constraint>
   <user-data-constraint id="UserDataConstraint_1">
      <transport-guarantee>NONE</transport-guarantee>
   </user-data-constraint>
</security-constraint>
<security-role id="SecurityRole_1">
   <description>All Authenticated Users Role.</description>
   <role-name>All Role</role-name>
</security-role>
```

web.xml – Form login

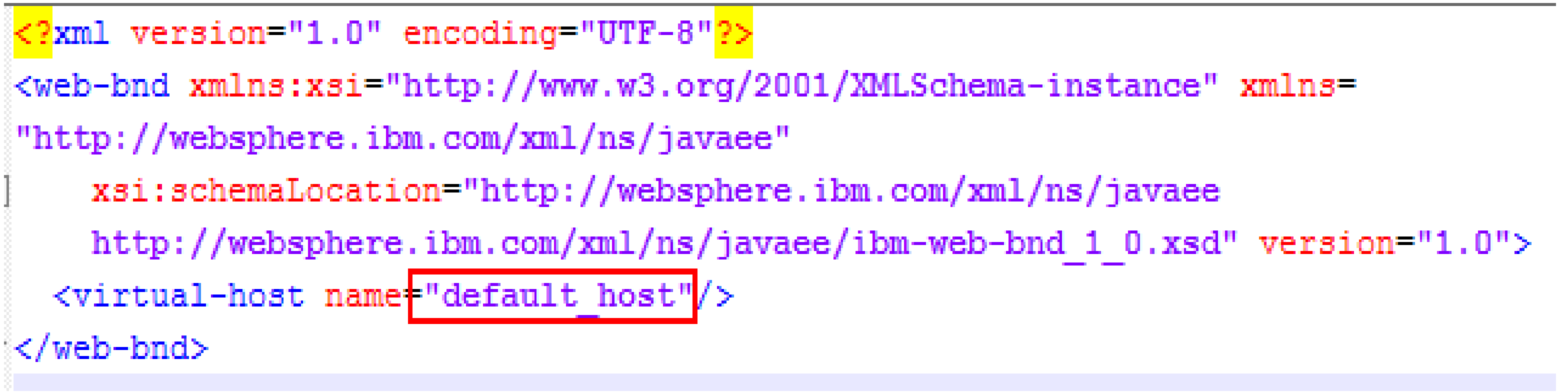
A security constraint is defined in the deployment descriptor (web.xml) that tells the server to send a form login to collect user data, verify that the user is authorized to access the application, and, if so, display the JSP page to the user.



```
<security-constraint>
        <display-name>SecurityConstraint</display-name>
       <web-resource-collection>
            <web-resource-name>WRCollection</web-resource-name>
           <url-pattern>/*</url-pattern>
    </web-resource-collection>
       <auth-constraint>
            <role-name>loginUser</role-name>
       </auth-constraint>
       <user-data-constraint>
           <transport-guarantee>NONE</transport-guarantee>
       </user-data-constraint>
  </security-constraint>
 <login-config>
       <auth-method>FORM</auth-method>
    <form-login-config>
            <form-login-page>/logon.jsp</form-login-page>
           <form-error-page>/logonError.jsp</form-error-page>
     </form-login-config>
<login-config>
<security-role>
   <role-name>loginUser</role-name>
</security-role>
```

ibm-web-bnd.xml

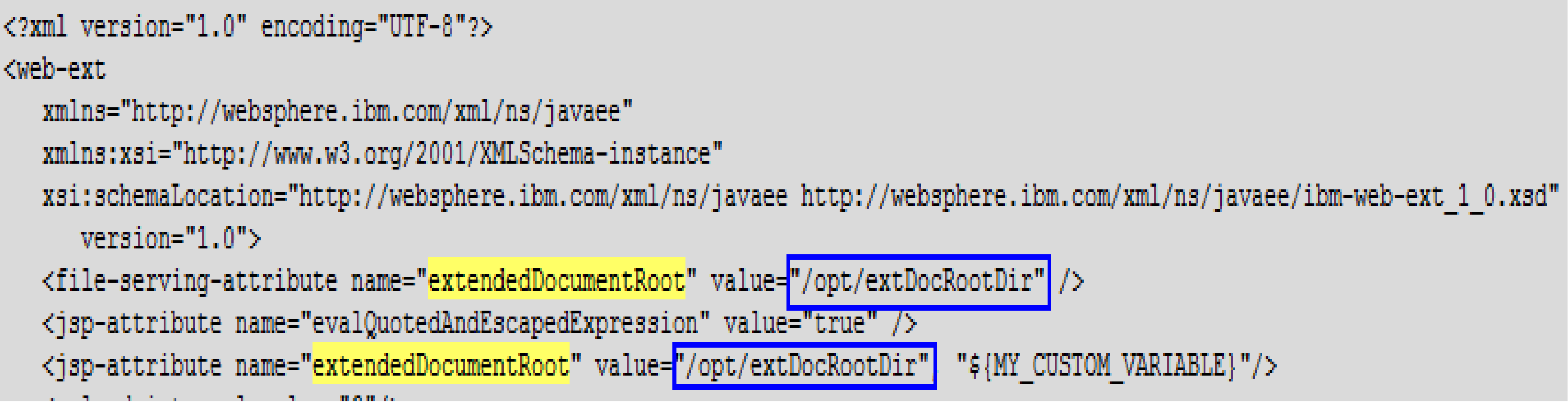
- The ibm-web-bnd.xml file contains bindings to references used at the Web module level as defined by IBM.
- The ibm-web-bnd.xml file shows what virtual host the application is mapping to. You can locate the ibm-web-bnd.xml file in the *{WAS_ROOT}/profiles/profilename/config/cells/cellname/applications/enterpriseAppName/deployments/deployedName/webModuleName/WEB-INF* directory.



```
<?xml version="1.0" encoding="UTF-8"?>
<web-bnd xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns=
"http://websphere.ibm.com/xml/ns/javaee"
    xsi:schemaLocation="http://websphere.ibm.com/xml/ns/javaee
    http://websphere.ibm.com/xml/ns/javaee/ibm-web-bnd_1_0.xsd" version="1.0">
  <virtual-host name="default_host"/>
</web-bnd>
```

ibm-web-ext.xml

- The ibm-web-ext.xml file is used by WebSphere Application Server to support additional options beyond the J2EE specification such as reloading intervals, JSP configuration parameter and Webcontainer attributes.
- This file can be located in the *{WAS_ROOT}/ profiles/config/cells/cellName/applications/enterpriseAppName/deployments/deployedName/webModuleName/WEB-INF* directory.
- Below is the sample of the ibm-web-ext.xml.



```
<?xml version="1.0" encoding="UTF-8"?>
<web-ext
   xmlns="http://websphere.ibm.com/xml/ns/javaee"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://websphere.ibm.com/xml/ns/javaee http://websphere.ibm.com/xml/ns/javaee/ibm-web-ext_1_0.xsd"
      version="1.0">
   <file-serving-attribute name="extendedDocumentRoot" value="/opt/extDocRootDir" />
   <jsp-attribute name="evalQuotedAndEscapedExpression" value="true" />
   <jsp-attribute name="extendedDocumentRoot" value="/opt/extDocRootDir"  "${MY_CUSTOM_VARIABLE}"/>
```

Summary

- Overview of Servlet specification
- Webcontainer settings
- Webcontainer attributes, what they are using for and how to configure them in the ibm-web-ext.xml file.
- Webcontainer custom properties, what they are using for and how to configure them in the administrative console.
- Virtual host configuration and how to map the web application to a virtual host.
- Java EE application and its XML configuration files, where to locate the application XML files and what to look for in the XML files.

References

- **Webcontainer settings** http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-express-iseries&topic=uweb_rcont
- **Webcontainer custom properties** http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-base-dist&topic=rweb_custom_props
- **Virtual hosts** http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-express-dist&topic=crun_vhost
- **Asynchronous request dispatcher** http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-express-iseries&topic=cwebard
- **Redbook: WebSphere Application Server V6: Webcontainer Problem Determination** http://www.redbooks.ibm.com/redpapers/pdfs/redp4058.pdf

Connect with us!

1. **Get notified on upcoming webcasts**
   Send an e-mail to wsehelp@us.ibm.com with subject line “wste subscribe” to get a list of mailing lists and to subscribe

2. **Tell us what you want to learn**
   Send us suggestions for future topics or improvements about our webcasts to wsehelp@us.ibm.com

Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at: http://www.ibm.com/software/websphere/support/supp_tech.html
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at: http://www.ibm.com/developerworks/websphere/community/
- Join the Global WebSphere Community: http://www.websphereusergroup.org
- Access key product show-me demos and tutorials by visiting IBM Education Assistant: http://www.ibm.com/software/info/education/assistant
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically: http://www.ibm.com/software/websphere/support/d2w.html
- Sign up to receive weekly technical My Notifications emails: http://www.ibm.com/software/support/einfo.html

Questions and Answers