BEA WebLogic Server [R] Application Container

Version: 10.3 Tech Preview

Document Date: October 2007

Table of Contents

Overview of Application Container

The following sections provide information on WebLogic Server (WLS) application container features for this release:

Version Independent Namespace

The following sections provide information on version independent namespaces.

Purpose

The purpose of this feature is to make WLS specific deployment descriptors easier to use and to reduce the work load of schemas upgrade.

Summary

There are two major problems from WLS current schemas' namespace convention:

1. Version information of WLS is included in the namespace.
2. All schemas are using the same namespace.

Version information is removed from namespace and each deployment descriptors schema has a separate namespace.

Versioned Namespace

The main problem with the current namespace is that it includes version information. When schema changes happen for a minor release or for a maintenance pack, BEA updates the namespace for that schema. This becomes a usability issue for application developers coding new descriptors with similar looking but different namespaces.

Namespace Sharing

The current convention also forces the same namespace on all the schemas. This contributes to the usability issue mentioned in the previous section. Also, it adds to maintenance by forcing BEA to move the unchanged schemas to a new release (every major release). This model doesn't fit well with modular development. As module producers grow in number, additional extra communication and co-ordination is required to avoid type name collisions for new types being added.

Solution

The BEA solution addresses the two major problems and lays out a road map for schema versioning in the future. The namespace will be broken into multiple namespaces, one per

descriptor. This promotes the schemas to indivitually and alleviates the problems mentioned above. Representing an entire namespace in one schema also provides a cleaner extensibility model for application developers and ISVs.

BEA also removed the version from namespace to make them more stable. The following new namespaces go forward, based on the pattern http://www.bea.com/ns/weblogic/<root element name> but schema owners may choose to have a different name if they like.

1. Application Container
   http://www.bea.com/ns/weblogic/weblogic-application
   http://www.bea.com/ns/weblogic/weblogic-extension
   http://www.bea.com/ns/weblogic/weblogic-application-client
2. EJB Container
   http://www.bea.com/ns/weblogic/weblogic-ejb-jar
   http://www.bea.com/ns/weblogic/weblogic-rdbms-jar
   http://www.bea.com/ns/weblogic/persistence-configuration
3. Interception
   http://www.bea.com/ns/weblogic/weblogic-interception
4. JDBC
   http://www.bea.com/ns/weblogic/jdbc-data-source
5. JMS
   http://www.bea.com/ns/weblogic/weblogic-jms
6. Connector
   http://www.bea.com/ns/weblogic/weblogic-connector
7. Web Container
   http://www.bea.com/ns/weblogic/weblogic-web-app
8. Web services
   http://www.bea.com/ns/weblogic/weblogic-webservices
   http://www.bea.com/ns/weblogic/weblogic-wsee-clientHandlerChain
   http://www.bea.com/ns/weblogic/webservice-policy-ref
   http://www.bea.com/ns/weblogic/weblogic-wsee-standaloneclient
9. Deployment
   http://www.bea.com/ns/weblogic/deployment-plan

Schema Versioning

All the schema changes that happen within a namespace would be backwards compatible. So any given descriptor from a previous version of the namespace would be parsed correctly with the latest schema. This is done using the “version” attribute of <schema> element at the top of each schema. For example:

```
<xsd:schema …
  version=”1.0”>
..
</xsd:schema>
```

An attribute “version” is also introduced for the root element of each descriptor. This is somewhat similar to the version attribute for each of the JavaEE descriptors. The JavaEE descriptors version attribute is required and is an enumeration of supported versions. But we don’t have to be that strict because our parsing infrastructure does a good job of munging the document into the latest schema structure anyways. The version attribute is optional and is of string type. The reason we might need this in the future to let any container force the user to put in a value to get a new feature working. Here is how this would look in a descriptor.

```
<weblogic-ejb-jar version=”1.0”>
</weblogic-ejb-jar>
```

Comprehensive Deployment Descriptor View

The following sections provide information on deployment descriptor view.

Purpose

AppMerge is required to parse all descriptors currently covered by SessionHelper for the JSR-88 Deployment View, because of the following:

1. It is cleaner design for the application container to do descriptor parsing rather than deployment sub-system doing so.
2. SessionHelper is modularized to limit dependencies on other containers due to the descriptors parsing function. AppMerge now performs all descriptors parsing functionality formerly done by SessionHelper.

Summary

Additional descriptors are supported in AppMerge for the deployment view.

The list of descriptors AppMerge supported prior to this release.

Descriptor Name	Descriptor URI	in Module/Flow
Webapp descriptor	WEB-INF/web.xml	WARModule
Connector descriptor	META-INF/ra.xml	RARModule
Application descriptor	META-INF/application.xml	ApplicationViewerFlow
EJB descriptor	META-INF/ejb-jar.xml	EJBModule

Webservice descriptor in Webapp	WEB-INF/webservices.xml	WARModule
Webservice descriptor in EJB	META-INF/webservices.xml	EJBModule
Weblogic Webapp descriptor	WEB-INF/weblogic.xml	WARModule
Weblogic EJB descriptor	META-INF/weblogic-ejb-jar.xml	EJBModule
Weblogic Application descriptor	META-INF/weblogic-application.xml	ApplicationViewerFlow
Weblogic Webservice descriptor in Webapp	WEB-INF/weblogic-webservices.xml	WARModule
Weblogic Webservice descriptor in EJB	META-INF/weblogic-webservices.xml	EJBModule
Persistence	META-INF/persistence.xml, META-INF/persistence-configuration.xml	ApplicationViewerFlow
Extension	META-INF/weblogic-extension.xml WEB-INF/weblogic-extension.xml	ApplicationViewerFlow

The following descriptors are added to AppMerge for this release:

Descriptor Name	**Descriptor URI**	**in Module/Flow**
Webservice descriptor in Webapp for 8.1	WEB-INF/web-services.xml	Need add Impl in WARModule
Webservice descriptor in EJB for 8.1	META-INF/web-services.xml	Need add Impl in EJBModule
Application client descriptor	META-INF/application-client.xml	Add impl in CARModule
Wldf-resource descriptor	META-INF/weblogic-diagnostics.xml	Need add Impl in EARModule and top level is highest priority
Weblogic Connector descriptor	META-INF/weblogic-ra.xml	Need add Impl in RARModule
Weblogic Application client	META-INF/weblogic-application-client.xml	Add impl in CARModule

descriptor		
Weblogic Webservice policy descriptor in Web	WEB-INF/weblogic-webservices-policy.xml	Need add impl in WARModule
Weblogic Webservice policy descriptor in EJB	META-INF/weblogic-webservices-policy.xml	Need add impl in EJBModule
JDBC	*-jdbc.xml	JDBCModule
JMS	*-jms.xml	JMSModule
WLS CMP	META-INF/weblogic-cmp-rdbms-jar.xml	EJBModule

External Diagnostics Descriptor

The following sections provide information on external diagnostic descriptors.

Purpose

J2EE container and AppMerge are required to support external diagnostic descriptors which may be provided by the end-user for deployment. This allows users to easily integrate the diagnostic function into an application which has not yet imported diagnostic descriptors.

Summary

The path of the external diagnostic descriptors is defined in plan.xml file as external entry.

This feature supports the deployment view and deployment of an application or a module detecting the presence of an external diagnostics descriptor if the descriptor is declared in the plan.

How to Configure External Diagnostics Descriptors

Define diagnostic descriptor as external and configure it's uri in plan.xml file, for example:

```
<module-override>
    <module-name>reviewService.ear</module-name>
    <module-type>ear</module-type>
    </module-descriptor>
    <module-descriptor external="true">
```

```
      <root-element>wldf-resource</root-element>
      <uri>META-INF/weblogic-diagnostics.xml</uri>
```

……

……

```
  </module-override>
  <config-root>D:\plan</config-root>
```

Then place the external diagnostic descriptor file under the uri, using the above example, place the descriptor file under D:\plan\ META-INF.

WebLogic Descriptor Modularization

The following sections provide information on WebLogic descriptor modularization.

Purpose

Currently in WebLogic Server release, the descriptor schema files (xsb) are duplicated in both weblogic.jar and jar files under lib/schema directory. This situation causes more maintenance work when modifying a descriptor.

Solution

WebLogic specific descriptor bean files and schema files are now shipped in com.bea.core.descriptor.wl_VERSION.jar and com.bea.core.descriptor.wl.binding_VERSION.jar under modules directory.