

IBM WebSphere Application Server Community Edition Version 3.0

-- More than Java EE 6

1

In this presentation, you will learn the major features provided in IBM WebSphere® Application Server Community Edition Version 3.0(CE 3.0 hereafter). CE 3.0 is a Java™ EE 6 certified application server, besides that, it also supports OSGi programming model.

Agenda

- Overview
- Feature highlights
- OSGi support
- Summary
- Q&A

The presentation includes overview, feature highlights, how OSGi programming model is supported and final wrap-up.

History

- Started in 2005, based on Apache Geronimo
- Mission
 - Create a free application server from Best-of-Breed Open Source components with world class IBM services
 - Comply with technical specifications for easy development
 - Provide migration paths to other products in WebSphere family

3

IBM WebSphere Community Edition server was firstly release in 2005 as version 1.0, which was based on Apache Geronimo 1.0 and complies with Java EE 5 specifications. As a member of WebSphere family, CE harvests best-of-breed open source components and has been released as a free product for development and use. Furthermore, CE provides migration paths to other products in WebSphere family, such as WebSphere Application Server, WebSphere Application Server Network Deployment.

Feature highlights



- Small Footprint: ~100MB download
- Liberal product license
 - Free to use in development, testing, and production
- Custom Server Assembly
 - Customizable server assemblies that contain only components you want or required by your applications
- Full featured administrative console
 - For almost all daily administrative operations
 - Easing administration, deployment, monitoring, and debugging!
- Eclipse plug-in with Helios support
 - For creating, deploying, and debugging applications
- Database Support
 - Integrated Apache Derby for development and production
 - IBM DB2®, MySQL, MS SQL Server, Oracle, and others

4

The footprint of a CE server is about 100 Megabytes. You can use CE server for testing, development or production without any license fee. CE server has a full featured web console for daily operations such as performance tuning, asset management and application deployment. With a fully modularized architecture, you can even extract a new server assembly that contains specific components that you need. CE server embeds Apache Derby as its database server and provides various support to different database vendors such as IBM DB2, MySQL, Oracle, MS SQL server and many more. For application development and debugging, you can use CE server adapter in Eclipse development tools directly.

Feature highlight – Custom Server Assembly

Full featured console - it's even in Chinese!

Select the functions you want - that's it!

Name	Version	Category
<input checked="" type="checkbox"/> Geronimo Plugin Group :: Framework	3.0-w20110721	Geronimo Plugin Group
<input type="checkbox"/> Geronimo Plugin Group :: Client	3.0-w20110721	Geronimo Plugin Group
<input type="checkbox"/> Geronimo Plugin Group :: Console Tomcat	3.0-w20110721	Geronimo Plugin Group
<input type="checkbox"/> Geronimo Plugin Group :: EBA Tomcat	3.0-w20110721	Geronimo Plugin Group
<input type="checkbox"/> Geronimo Plugin Group :: EJB	3.0-w20110721	Geronimo Plugin Group
<input type="checkbox"/> Geronimo Plugin Group :: JMS (ActiveMQ)	3.0-w20110721	Geronimo Plugin Group
<input type="checkbox"/> Geronimo Plugin Group :: Java EE 6 Tomcat	3.0-w20110721	Geronimo Plugin Group
<input type="checkbox"/> Geronimo Plugin Group :: Java EE 6 Web Profile Tomcat	3.0-w20110721	Geronimo Plugin Group
<input type="checkbox"/> Geronimo Plugin Group :: Persistence	3.0-w20110721	Geronimo Plugin Group
<input type="checkbox"/> Geronimo Plugin Group :: Web Tomcat	3.0-w20110721	Geronimo Plugin Group
<input type="checkbox"/> Geronimo Plugin Group :: Web Services Axis2	3.0-w20110721	Geronimo Plugin Group

To customize a new server assembly, you can use CE web console by selecting the functions you want and then follow the wizard. The only thing worth noting is that you must select Geronimo Framework plug-in group for each new server assembly.

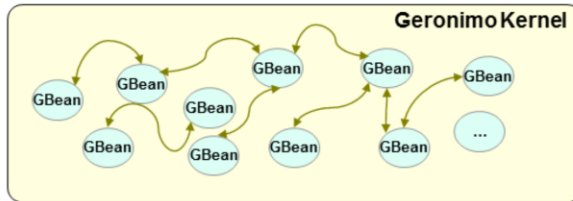
Behind the scene – An open source party?



6

As the foundation of a CE server, Apache Geronimo integrates lots of open source projects such as OpenEJB, Tomcat, OpenJPA, ActiveMQ, Derby, Tranql, Tuscany and so on. For example, Tomcat is the Web container for both CE server and Geronimo.

At the core – Geronimo kernel



Geronimo Kernel glues third party components into an application server, wrapping each of them as a GBean.

What is GBean?

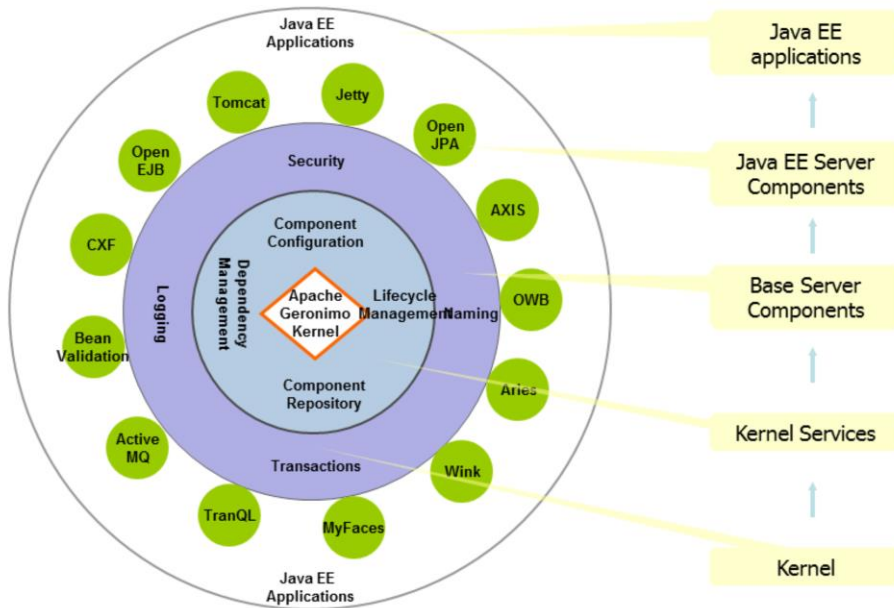
- POJO wired with xml configuration or annotation

- May implement some lifecycle interfaces

- Managed by Geronimo kernel

Geronimo kernel glues all those open source components into an application server using its GBean techniques. GBean is a POJO wired with xml file configuration or annotations, the life cycle of GBean is managed by Geronimo kernel.

Geronimo architecture



Looking at the architecture of Geronimo, the kernel provides kernel services for various components, and each component implements Java EE component specifications. By binding them together, you have a fully compliant Java EE application server.

Support service & more

- IBM provides paid support service for Geronimo.
- IBM also provides a free commercial distribution of Geronimo, called WebSphere Application Server Community Edition.
 - 3 tiers of support contracts available - Very low cost
 - Long lived service/support streams
 - Quick availability of security fixes and hi impact bugs
 - No need to maintain a build environment and support stream. IBM does this for you.
 - Access to experts who are members of the OS development communities
 - Integration with WebSphere family of products
 - Redistributable (need to sign a simple agreement with IBM)



IBM provides paid support service for both Geronimo and CE server. As for CE server, IBM provides 3 tiers of support at very low cost for each version and integration with other WebSphere products. If you want to redistribute CE server, you need to sign a simple agreement with IBM.

Agenda

- CE overview
- CE V3.0 highlight
- Support for OSGi Enterprise
- Summary

Now, let's take a look at some major new features in CE 3.0.

What's new in V3.0

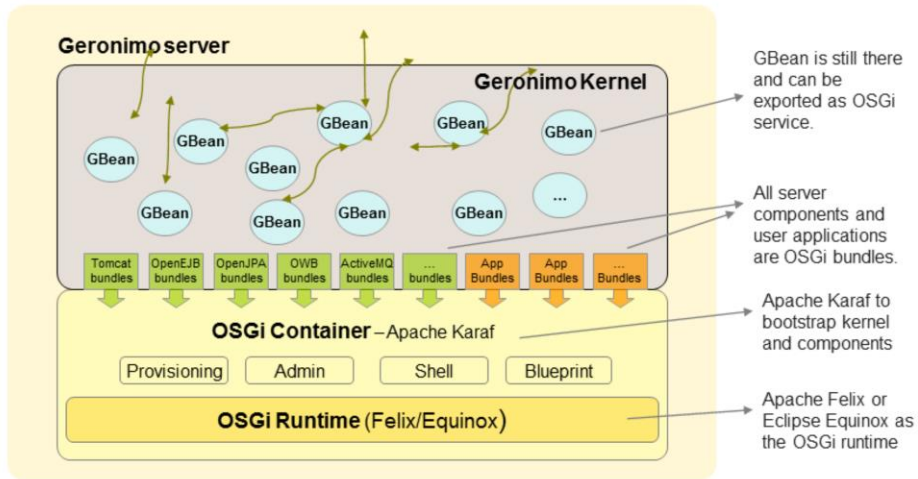
- Java EE 6 support
- Kernel OSGi-fication
- OSGi basic programming model support
- OSGi Enterprise support
- Development tool empowered for OSGi
- Improved user experience

More than Java EE 6

CE 3.0 re-factored its kernel based on OSGi framework and thus provides a hybrid run time environment for both Java EE and OSGi applications. CE 3.0 is fully compliant with Java EE 6 specifications and provides enterprise support for OSGi applications. You can use CE 3.0 for Java EE application development and an additional OSGi development tool for OSGi development.

Kernel OSGi-fication (1 of 2)

- CE 3.0 is based on Geronimo 3.0, which brings OSGi into its kernel .



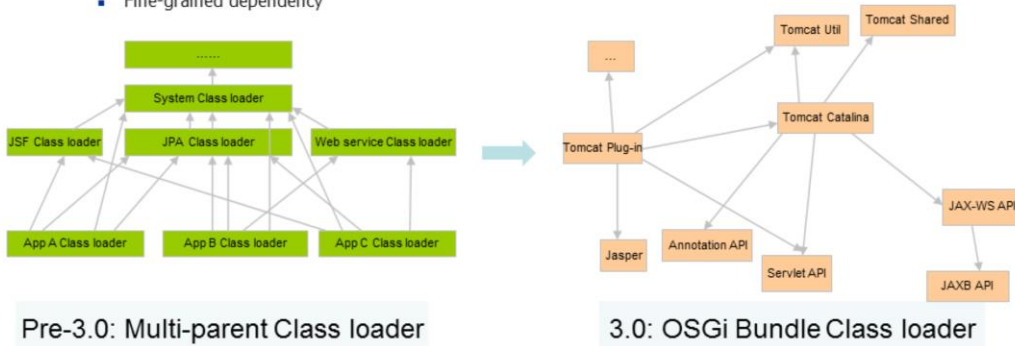
13

As for the Geronimo kernel in 3.0, GBeans are still there and they can be exported as OSGi services. All server components and user applications will be managed as OSGi bundles. Apache Karaf is the OSGi container to bootstrap Geronimo kernel and other components. You can choose to use Apache Felix or Eclipse Equinox as your OSGi run time environment cause both modules are embedded in CE server.

Kernel OSGi-fication (2 of 2)

Class loader rewritten to better manage dependencies

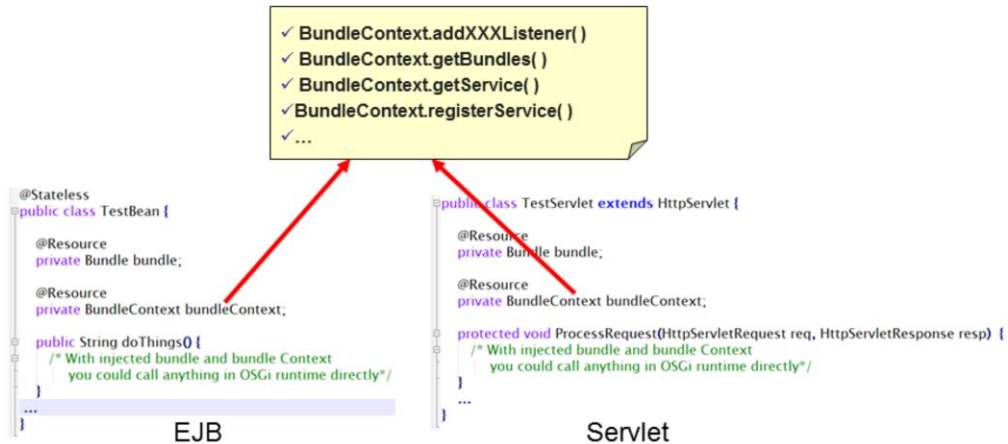
- **Pre-3.0 : Multi-parent class loader**
 - One class loader can have more than one parent class loaders.
 - Modules are wired to each other with explicit dependency configuration at module level.
 - Having potential class duplicates/conflicts and class loading performance challenges
- **3.0 : OSGi Bundle Class loader**
 - One class loader for One Bundle/Jar
 - Modules are wired to each other at Java package level in OSGi runtime.
 - Fine-grained dependency



The advantage of class loading mechanism in CE 3.0 is we use OSGi bundle class loader instead of multi-parent class loader. By using OSGi bundle class loader, each bundle has one class loader and modules are wired to each other at package level. While in multi-parent class loader scenarios, modules are wired to each other with explicit dependency configuration at module level which might lead to certain class casting exceptions and performance issues.

OSGi basic programming model support

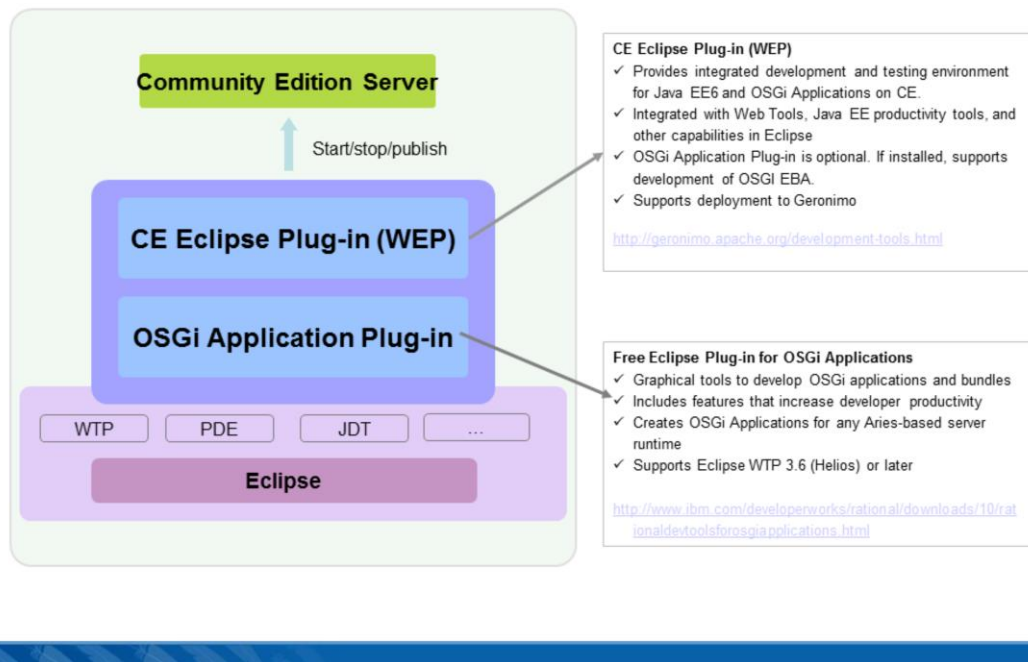
- Bundle and BundleContext can be injected into Java EE components using the @Resource annotation, opening the door to OSGi world...



15

As for the support of OSGi programming model, we can use @Resource annotation to inject OSGi bundle and bundle context into Java EE components, thus you can access any resources within a OSGi bundle from an Java EE component such as EJB and servlet.

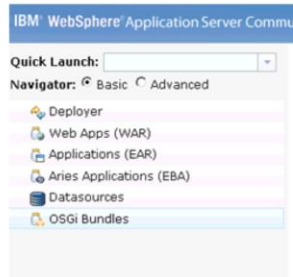
Development tool empowered for OSGi



There are two different tools you can use in Eclipse. CE Eclipse plug-in provides integrated development and testing environment for Java EE 6 and OSGi applications. OSGi application plug-in provides supports for creating OSGi application and bundles. By using both tools together, we can development and test OSGi enterprise applications on CE 3.0 server.

Improved user experience

- Administrative console refactored
 - Enhanced navigation experience with Web 2.0 technology and user engineering
 - Integrated OSGi bundle management console
- 3rd party component configuration eased
 - Use 3rd-components' original configuration style



The screenshot shows the 'OSGi Bundles' management console. It includes a search bar for 'Find Packages' and a table of installed bundles. The table has columns for Id, Symbolic Name, Version, Type, State, and Utilities. The bundles listed are configuration bundles for collector-tool-agent, wink-deployer, wink, udi-tomcat, and udi-db.

Id	Symbolic Name	Version	Type	State	Utilities
368	com.ibm.wasce.configs.collector-tool-agent-config	0.0.0.3_0_0_0	CONFIGURATION	ACTIVE	
366	org.apache.geronimo.configs.wink-deployer	3.0.0.w20110721	CONFIGURATION	ACTIVE	
362	org.apache.geronimo.configs.wink	3.0.0.w20110721	CONFIGURATION	ACTIVE	
361	org.apache.geronimo.configs.udi-tomcat	3.0.0.w20110721	CONFIGURATION	ACTIVE	
360	org.apache.geronimo.plugins.udi-db	3.0.0.w20110721	CONFIGURATION	ACTIVE	

17

In CE 3.0, the navigation panel is redesigned as two new views. For frequently used tasks, you can use basic views and you can use advanced view for full operations to the server. You can also use the quick launch bar to locate a menu by typing keywords. And there is an integrated OSGi management portlet on the console for bundle management.

Agenda

- CE overview
- CE 3.0 highlight
- Support for OSGi Enterprise
- Summary

In the following pages, we'll learn about some basics regarding OSGi enterprise support.

What is it – OSGi Enterprise Spec V4.2

- Release on March 22nd, 2010
 - Made by the OSGi Enterprise Expert Group (EEG)
- Brings Enterprise technologies and OSGi together
- Using existing Java SE/EE specifications:
 - JNDI, JPA, JTA, JMX, WebApps...
- Adds Spring-derived component model and dependency injection container – Blueprint Container
- Java EE provides the core enterprise application programming model.
- **Deploying modules as OSGi bundles simplifies reuse between applications, provides versioning, encourages (and enforces) modular design and enables dynamic module updates.**



OSGi Enterprise specification was released by OSGi Enterprise Expert Group(EEG) on March 22, 2010. The specification intends to integrate existing enterprise technologies such as JNDI, persistence, transaction support into OSGi world. For example, the blueprint container specification is used for dependency injection which is derived from Spring framework.

Implementation and extension by



Apache Aries, a new Apache project for OSGi enterprise support

- Creating components to enable OSGi Application Programming Model
- Implementing OSGi EEG standards
 - Blueprint container
 - JNDI integration
 - JPA integration
 - JTA integration
 - JMX
 - META-INF/services handler
 - ...
- Introducing new technologies to push into standards
 - Enterprise Bundle Application: Programming Model for developing, assembling and deploying, as OSGi bundles, modular applications that exploit Java EE and OSGi technologies.

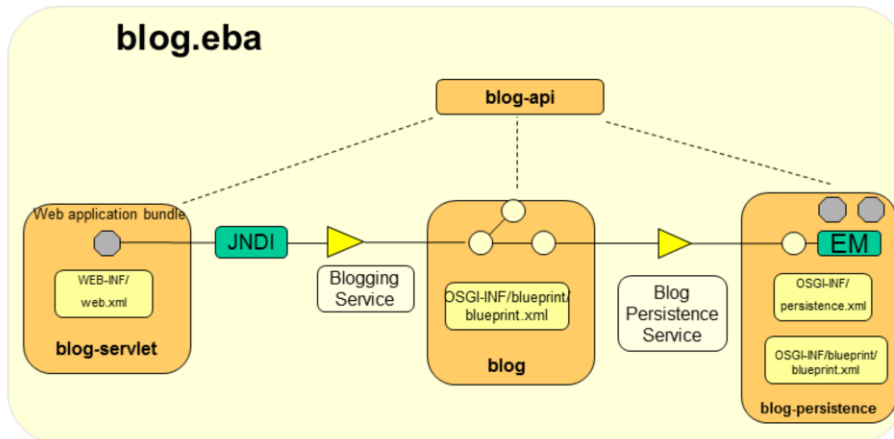
Aries is currently used by:

- Geronimo
- JBoss OSGi
- WebSphere Application Server Feature Pack for OSGi Applications and JPA 2.0

Apache Aries becomes a top-level Apache project in 2010 and delivers a set of pluggable Java components for enabling enterprise OSGi application programming model. The Aries project provides enterprise OSGi components such as blueprint container and projects to integrate enterprise technologies such as JNDI, JTA, JPA into OSGi environment to make these available to OSGi bundles.

Example – A “Blog” Application

- Blueprint Component Model
- JNDI integration
- Container-Managed JPA Persistence & JTA Transactions
- Application Assembly & Deployment

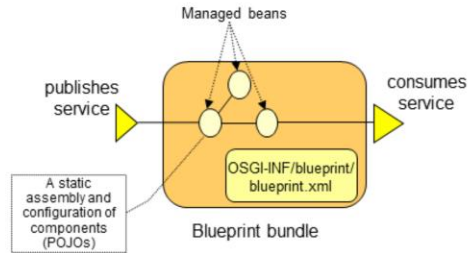


21

The Blog sample is used to demonstrate how to use JPA, JTA and blueprint in an enterprise OSGi application. The blog enterprise bundle application is composed by four bundles: blog-api bundle, blog web bundle, blog persistent bundle and blog business bundle. The blog persistent bundle uses JPA as the persistence layer.

Aries Blueprint Container

- A DI container standardizing established Spring conventions
- XML Blueprint definition describes component configuration and scope
 - Optionally publish and consume components to/from OSGi service registry
- Simplifies unit testing outside either Java EE or OSGi runtime



Sample blueprint.xml

```

.....
<bean id="blogEntryManager" class="org.apache.aries.samples.blog.biz.BlogEntryManagerImpl">
  <property name="persistenceService" ref="persistenceManager"/>
</bean>

<bean id="bloggingServiceComponent" class="org.apache.aries.samples.blog.biz.BloggingServiceImpl">
  <property name="blogEntryManager" ref="blogEntryManager"/>
  <property name="blogAuthorManager" ref="blogAuthorManager"/>
</bean>

<service ref="bloggingServiceComponent" interface="org.apache.aries.samples.blog.api.BloggingService"/>

<reference id="persistenceManager" interface="org.apache.aries.samples.blog.api.persistence.BlogPersistenceService"/>

```

22

The Blueprint Container specification uses an extender pattern, whereby an extender bundle monitors the state of bundles in the framework and performs actions on behalf of those bundles based on their state. The Blueprint extender bundle waits for the bundles to be activated and checks whether they are Blueprint bundles. A bundle is considered to be a Blueprint bundle when it contains one or more Blueprint XML files. These XML files are at a fixed location under the OSGI-INF/blueprint/ directory or are specified explicitly in the Bundle-Blueprint manifest header. During initialization, the Blueprint Container ensures that mandatory service references are satisfied, registers all the services into the service registry, and creates initial component instances. The Blueprint extender bundle also destroys the Blueprint Container for a bundle when the bundle is stopped.

Aries JNDI integration

- Provides JNDI-based access to OSGi Service Registry

```
<blueprint xmlns=...>
  <bean id="bloggingServiceComponent"
        class="org.apache.aries.samples.blog.biz.BloggingServiceImpl">
  </bean>
  <service ref="bloggingServiceComponent"
          interface="org.apache.aries.samples.blog.api.BloggingService"/>
  ...
</blueprint>
```

registerService

OSGi
Service Registry

getService

JNDI Context

- A way for a Web component to access a Blueprint component

```
InitialContext ic = new InitialContext();
BloggingService blog= ic.lookup("osgi:services/"
    + BloggingService.class.getName());
```

Aries JNDI integration provides JNDI-based access to OSGi service registry. You can use blueprint to register OSGi services into the OSGi service Registry. To look up OSGi services, use `osgi:services:` scheme to locate the service you are looking for.

Aries JPA Container – Blueprint Integration

- The Aries JPA container context bundle provides a blueprint namespace for dependency injection of managed JPA resources.
- Managed persistence units (EntityManagerFactory objects) can be injected with or without a JTA Transaction Services implementation.
- Managed persistence contexts (EntityManager objects) are only available with a JTA Transaction Services implementation.
- Both managed persistence units and managed persistence contexts behave as per the JPA specification.

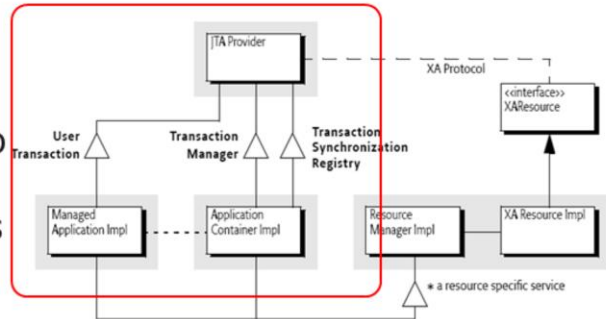
```
<bean id="persistenceImpl"  
  class="org.apache.aries.samples.blog.persistence.jpa.BlogPersistenceServiceImpl">  
  <tx:transaction method="" value="Required" />  
  <jpa:context property="entityManager" unitname="blogExample" />  
</bean>  
  
<service ref="persistenceImpl"  
  interface="org.apache.aries.samples.blog.api.persistence.BlogPersistenceService">  
</service>
```

Sample blueprint.xml with JPA resource injection and container-managed transactions

Aries JPA provides managed persistence for the blueprint container. The Aries JPA container provides JPA support for OSGi persistence bundles. In the OSGi persistence bundle, both managed persistence units and managed persistence contexts behave per the JPA specification. The code snippet on this page is a sample blueprint file with JPA resource injection and container-managed transaction.

Aries JTA integration

- Apache Aries integrates the OSGi Transaction Service Reference Implementation (Apache Geronimo Transaction Manager).
- Transaction-blueprint bundle registers a "jta" Blueprint namespace handler to provide container-managed transactions for BP components.

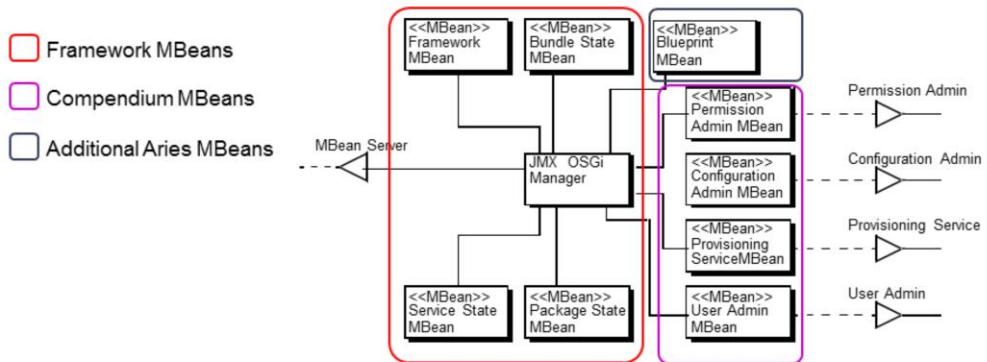


25

Aries transaction support integrates the OSGi transaction service reference implementation which also serves as Geronimo transaction manager. You can register a JTA blueprint namespace handler to provide container-manager transaction.

Aries JMX Integration

- Implements OSGi JMX specification.
- Aries JMX bundle automatically registers the JMX MBeans into any `javax.management.MBeanServer` service in the OSGi Service Registry.

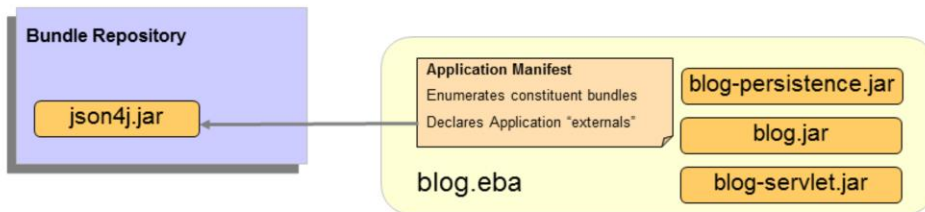


26

Aries JMX bundle automatically registers the JMX MBeans into any `javax.management.MBeanServer` service in the OSGi Service Registry.

Aries Application Assembly and Deployment – EBA

- The “application” project provides pluggable infrastructure to support deployment of an application consisting of a collection of bundles as a logical unit from an Enterprise Bundle Archive (archive with .eba extn).
 - An “Enterprise OSGi Application”.
- Constituent bundles may be contained (“by-value”) in the .eba archive or referenced in APPLICATION.MF
- Only explicitly declared Services are exposed from the application.
- Configuration by exception - absence of APPLICATION.MF means:
 - Application content is the set of bundles contained by-value plus any repository-hosted dependencies identified during deployment.



27

The suffix of an Aries application is EBA, which means enterprise bundle archive. The EBA must have an application.mf manifest file in it. Only explicitly declared services are exposed from the application.

Agenda

- CE overview
- CE3.0 highlight
- Support for OSGi Enterprise
- Summary

Let's wrap-up the presentation.

Summary

- IBM CE server is a lightweight, customizable server environment that brings together many best-of-breed open source components.
- CE 3.0 is featured with Java EE 6 and OSGi programming model support, bringing all the benefits of OSGi into the EE world.
- CE is release by IBM and free-of-charge

To summarize the presentation, we need to remember that IBM WebSphere Community server is released by IBM and free to use. It's a light-weight, customizable server that brings together many best-of-breed open source components. The version 3.0 of CE server is feature with Java EE 6 and OSGi programming model support, and brings all the benefits of OSGi architecture into the Java EE world.

Resources

- Apache Geronimo Project
 - <http://geronimo.apache.org/>
- Apache Geronimo Documentation
 - <https://cwiki.apache.org/geronimo/>
- Apache Aries Project
 - <http://incubator.apache.org/aries/>
- WebSphere Application Server Community Edition
 - <http://www.ibm.com/websphere/wasce>

You may refer to the resources listed on this page to learn more about Apache Geronimo and CE.

Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, DB2, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

Java, and all Java-based trademarks and logos are trademarks of Oracle and/or its affiliates.

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2011. All rights reserved.