Red Hat JBoss Enterprise Application Platform 7.2

7.2.0 Release Notes

For Use with Red Hat JBoss Enterprise Application Platform 7.2

Last Updated: 2019-09-26

Red Hat JBoss Enterprise Application Platform 7.2 7.2.0 Release Notes

For Use with Red Hat JBoss Enterprise Application Platform 7.2

Legal Notice

Abstract

These release notes contain important information related to Red Hat JBoss Enterprise Application Platform 7.2.

Table of Contents

CHAPTER 1. SUPPORTED CONFIGURATIONS

The following configurations are newly supported for JBoss EAP 7.2.

Java Virtual Machine

- OpenJDK 8 for Windows Server
- OpenJDK 11 for Windows Server and Red Hat Enterprise Linux [1]
- Oracle JDK 11 [1]

[1] JDK 11 support does not include deploying JPMS (Java Platform Module System) user modules to the server.

Databases and Database Connectors

- IBM DB2 e11.1
- PostgreSQL 10.1
- EnterpriseDB Postgres Plus Advanced Server 10.1

JMS Providers and Adapters

- IBM MQ 9

Red Hat Developer Studio

- JBoss EAP 7.2 is certified for use with Red Hat Developer Studio 12.

See the Red Hat JBoss Enterprise Application Platform (EAP) 7 Supported Configurations page for full supported configuration details for JBoss EAP 7.2.

CHAPTER 2. NEW FEATURES AND ENHANCEMENTS

2.1. JAVA EE 8

Java EE 8 Support

JBoss EAP 7.2 includes support for the following Java EE 8 standards:

- JSR 250: Common Annotations 1.3
- JSR 338: JPA 2.2
- JSR 365: CDI 2.0
- JSR 367: JSON-B 1.0
- JSR 369: Servlet 4.0
- JSR 370: JAX-RS 2.1
- JSR 372: JSF 2.3
- JSR 374: JSON-P 1.1
- JSR 380: Bean Validation 2.0
- JSR 919: JavaMail 1.6

Java EE 8 Security Support

Java EE 8 includes support for JSR 375, which defines portable, plug-in interfaces for authentication and identity stores, and a new **injectable-type SecurityContext** interface that provides an access point for programmatic security. You can use the built-in implementations of these APIs, or define custom implementations.

JBoss EAP 7.2 now supports JSR 375.

2.2. SECURITY

Securing the JBoss EAP Management Console with Red Hat Single Sign-On

You can now configure Red Hat Single Sign-On to manage authentication for the JBoss EAP management console.

For more information, see Secure the Management Console with Red Hat Single Sign-On in *How to Configure Server Security*.

Server Blocks Non-SSL IIOP Socket When SSL Is Required

When the **server-requires-ssl** attribute is set to **true** in the IIOP subsystem, the server will now block attempts to connect to the non-SSL socket.

FIPS 140-2 Compliant Cryptography Using the BouncyCastle Providers

You can use the BouncyCastle providers to configure a FIPS compliant instance of JBoss EAP using the **elytron** subsystem. Full instructions are available at Enable FIPS 140-2 Cryptography for SSL/TLS Using BouncyCastle in *How to Configure Server Security* .

Define a FIPS 140-2 Compliant Credential Store Using the BouncyCastle Providers

You can use the BouncyCastle providers to obtain a FIPS compliant credential store. These credential stores can be defined using either of the following methods.

- For instructions on defining the credential store directly through the **elytron** subsystem, see Define a FIPS 140-2 Compliant Credential Store Using the BouncyCastle Providers .
- For instructions on defining the credential store offline using the WildFly Elytron tool, see Create and Modify Credential Stores Offline with the WildFly Elytron Tool .

Enable SASL Authentication for the Management Interfaces Using the CLI Security Commands

SASL authentication can now be enabled for the management interfaces using the **security enable-sasl-management** CLI command. This command creates all of the non-existing resources necessary to configure authentication.

For more information, see Enable SASL Authentication for the Management Interfaces Using the CLI Security Command in *How to Configure Server Security* .

Enable HTTP Authentication Using the CLI Security Commands

HTTP authentication can now be enabled for the Undertow security domain and the management interfaces using the **security** CLI commands.

- For the Undertow security domain, use the **security enable-http-auth-http-server** CLI command.
- For the management interfaces, use the **security enable-http-auth-management** CLI command.

For more information, see Enable HTTP Authentication for Applications Using the CLI Security Command and Enable HTTP Authentication for the Management Interfaces Using the CLI Security Command in *How to Configure Server Security* .

2.3. SERVER MANAGEMENT

Using Git to Manage Configuration Data

You can now use Git to manage and persist your server configuration data, properties files, and deployments. This not only allows you to manage the version history, but it also allows you to share server and application configurations across multiple servers and nodes using one or more Git repositories. This feature only works for standalone servers that use the default configuration directory layout.

For more information, see Using Git to Manage Configuration Data in the *Configuration Guide*.

Kill Servers in a Server Group

The **kill-servers** operation is now available for server groups in a managed domain. This is useful in cases where a problem is causing all servers in a server group to hang, so that you can kill all of the server processes in one operation as opposed to performing the **kill** operation on each server.

2.4. MANAGEMENT CLI

Keyboard Navigation Shortcuts

The management CLI now supports several ways to navigate around when editing a management CLI command. The keyboard shortcuts to use depend on which platform you are using. See Use Keyboard Navigation Shortcuts in the *Management CLI Guide* for the list of supported shortcuts.

Generate Output for HTTP Management API

The **echo-dmr** command provides a new **--compact** argument to display content on a single line. When used with the **--output-json** management CLI startup argument, this argument allows you to generate output that can be directly consumed by the HTTP Management API.

Management CLI Output Scrolling

The management CLI now supports scrolling directly inside the console if the output is longer than the terminal window. You can use the scroll wheel, directional arrows, or the **PgUp**, **PgDn**, **Home** and **End** keys to navigate through the output.

On Windows this feature is only available beginning with Windows Server 2016. There are no issues with other operating systems.

Searching Management CLI Output

You can now search multi-page output in the management CLI. See Searching Multi-page Output in the *Management CLI Guide* for more information.

Printing CLI Output in Color

You can now configure the management CLI to print the CLI log output in color based on the log message output type. For more information about the available colors and how to enable and disable color printing, see Configuring the Management CLI in the *Management CLI Guide*.

Enhanced Help

The management CLI **help** functionality has been updated to provide easier access to help information. The **help** command now features tab completion and can also show help information for management CLI operations and command actions.

See the *Management CLI Guide* for more information on using the management CLI **help** command.

Indicator for Required Attributes

When using tab completion in the management CLI, attributes that are required for the current operation are marked with a * character.

```
/subsystem=naming/binding=test:add( [TAB]
!              class          module
binding-type*  environment    type
cache          lookup         value
```

In the above example, pressing Tab after entering /**subsystem=naming/binding=test:add(** lists the available attributes and indicates that **binding-type** is a required attribute for this operation.

Viewing Multi-page Output

When you run the management CLI in interactive mode and the operation results in multiple pages of output, the command processor pauses the screen at the end of the first page. This allows you to page through the output one line or page at a time. The occurrence of multiple pages of output is indicated by a line of text displaying **--More(*NNN*%)--** at the end of the output.

See the *Management CLI Guide* for the options available if you encounter multiple page output when running a management CLI command.

Using for-done Control Flow

You can use **for-done** control flow in the management CLI to iterate over a collection returned from an operation and execute commands on each item in the collection.

For more information, see Use for-done Control Flow in the *Management CLI Guide*.

Output Operation Responses in JSON Format

You can configure the management CLI to output operation responses in pure JSON format by setting the **output-json** element to **true** in the ***EAP_HOME*/bin/jboss-cli.xml** file or by passing the **--output-json** flag in when starting the management CLI. By default, operation responses are displayed in DMR format.

Redirecting Output

Instead of printing output from a management CLI operation to the terminal, you can redirect the output using the following operators:

- **>**: Write output to a file on the file system.
- **>>**: Append output to a file on the file system.
- |: Redirect output to the **grep** command for searching the output.

For more information, see Redirect Output in the *Management CLI Guide*.

Unified Deployment Command

The management CLI **deployment** command allows you to manage your deployments using a unified interface to deploy, undeploy, enable, disable or list information about the deployments.

For more information, see Deploy an Application to a Standalone Server Using the Management CLI and Deploy an Application in a Managed Domain Using the Management CLI in the *Configuration Guide*.

2.5. MANAGEMENT CONSOLE

Topology View

In a managed domain, you can now see an overview of the hosts, server groups, and servers in the domain, and the status of each server. This is available from the **Runtime** tab by selecting **Topology**.

Breadcrumb Bar

When viewing resources, a breadcrumb bar is available at the top that allows you to easily switch between resources. From the breadcrumb bar, you can also open the resource in a separate window or switch to expert mode to browse the management model.

Navigation Enhancements

This release introduces a new interface for navigating JBoss EAP resources. You can use the arrow keys to navigate through the resource finder, pin frequently used items to stay at the top of the list, filter to quickly find items, and view the main attributes of a resource from its preview.

Deployment Enhancements

This release adds more support for deploying and managing your applications through the management console. You can drag and drop to add or replace deployments, browse deployment content to preview text and images, download deployments, and create exploded deployments.

Enable SSL Wizard for the Management Console

A wizard is now available to help you enable SSL for the HTTP management interface, which is used by the management console. Using the wizard, you can optionally create a truststore for mutual authentication as well as choose from the following keystore scenarios:

- You want to create a certificate store and generate a self-signed certificate.
- You already have the certificate store on the file system, but no keystore configuration.
- You already have a keystore configuration that uses a valid certificate store.

To access the wizard for a standalone server, select the **Runtime** tab, click **View** on the appropriate server, select **HTTP Management Interface** and click the **Enable SSL** button.

To access the wizard for a managed domain, select the **Runtime** tab, click **Hosts** and select the appropriate host, select **View → Management Interface → HTTP** and click the **Enable SSL** button.

Enable SSL Wizard for Undertow HTTPS Listeners

A wizard is now available to help you enable SSL for Undertow HTTPS listeners. Using the wizard, you can optionally create a truststore for mutual authentication as well as choose from the following keystore scenarios:

- You want to create a certificate store and generate a self-signed certificate *(not available in a managed domain)*.
- You already have the certificate store on the file system, but no keystore configuration.
- You already have a keystore configuration that uses a valid certificate store.

To access the wizard for a standalone server, click the **Configuration** tab, select **Subsystems → Web (Undertow) → Server**, click **View** on the appropriate server, select **Listener → HTTPS Listener**, select the appropriate HTTPS listener and click the **Enable SSL** button.

To access the wizard for a managed domain server, click the **Configuration** tab, click **Profiles** and select the appropriate profile, select **Web (Undertow) → Server**, click **View** on the appropriate server, select **Listener → HTTPS Listener**, select the appropriate HTTPS listener and click the **Enable SSL** button.

Configuring Logging Profiles

You can now use the management console to configure logging profiles in the **logging** subsystem.

Configuring the Security Manager Subsystem

It is now supported to configure the **security-manager** subsystem from the management console.

Configuring Elytron Components

It is now supported to configure the following Elytron components using the management console:

- Mapped role mappers
- Certificate authority accounts
- Custom security event listeners

It is also supported to create, deactivate, update, and change the account key for certificate authority accounts from the **Runtime** tab of the management console.

Viewing Session Details

You can now view detailed session information for deployments using the management console.

From the **Runtime** tab, choose the appropriate server, select **Web (Undertow) → Deployment**, choose the deployment, and click **View**. The **Sessions** page provides a table that lists the ID, creation time, and last accessed time for all active sessions. You can also click on a session to view its attributes.

Invalidating Active Sessions

You can now invalidate active sessions for deployments using the management console.

From the **Runtime** tab, choose the appropriate server, select **Web(Undertow) → Deployment**, choose the deployment, and click **View**. From the **Sessions** page, select the session ID to invalidate, and click **Invalidate session**. Click **Yes** on the popup that appears to invalidate the session.

Configuring Scattered Caches

This release introduces scattered caches, which can be configured from within the cache configuration in the **infinispan** subsystem.

In addition, scattered caches support the use of the **hotrod** cache store.

Additional Subsystem Configuration

The following subsystems have been added or enhanced to include additional configuration options, available from the **Configuration** tab:

- MicroProfile Config SmallRye
- EJB
- Infinispan
- JGroups
- JMX
- Messaging (ActiveMQ)
- Resource Adapters
- Security (Legacy)
- Web (Undertow)

Additional Subsystem Monitoring Support

This release provides new and enhanced monitoring support for the following subsystems, available from the **Runtime** tab:

- Batch (JBeret)
- Datasources
- JNDI
- EJB
- IO
- JAX-RS
- Messaging (ActiveMQ)
- Transaction
- Web (Undertow)
- Webservices

2.6. WEB SERVER

Undertow Byte Buffer Pools

You can now use Undertow byte buffer pools to allocate pooled NIO **ByteBuffer** instances. All listeners have a byte buffer pool and you can use different buffer pools and workers for each listener. Byte buffer pools can be shared between different server instances.

For more information, see Configuring Byte Buffer Pools in the *Configuration Guide*.

Setting the Default Cookie Version

Undertow now provides a way to set the default cookie version to use for cookies created by the application. For information about the new **default-cookie-version** attribute, see **servlet-container** Attributes in the *Configuration Guide*.

Allowing Unescaped Characters in a URL

You can now configure Undertow to allow non-escaped characters in a URL by setting the **allow-unescaped-characters-in-url** attribute for the HTTP, HTTPS, and AJP listeners. When this attribute is set to **true**, the listener processes any URL containing non-escaped, non-ASCII characters. When set to **false**, the listener rejects any URL containing non-escaped, non-ASCII characters with an **HTTP Bad Request 400** response code.

For more information about listener attributes, see Undertow Server Attributes in the *Configuration Guide*.

PROXY Protocol

Undertow now supports the PROXY protocol Version 1, as defined by The PROXY protocol Versions 1 & 2 specification. This option is disabled by default and must only be enabled for listeners that are behind a load balancer that supports the same protocol. It is configured using the new **proxy-protocol** attribute on the Undertow HTTP and HTTPS listeners.

For more information about listener attributes, see Undertow Server Attributes in the *Configuration Guide*.

Forwarded HTTP Extension

JBoss EAP 7.2 introduces the **Forwarded** handler, which implements RFC 7239, allowing servers behind a reverse proxy to receive peer and local addresses within the header.

Typically, this handler should not be used in conjunction with any of the **X-Forwarded-*** headers enabled on the reverse proxy. This means that you should either use this handler or enable the **proxy-address-forwarding** attribute in Undertow listeners.

Session Manager Operations

The following operations to get detailed session information are now available from the management CLI at /**deployment=*DEPLOYMENT_NAME*/subsystem=undertow**.

- **get-session-attribute**: Return a specific attribute for a session.
- **get-session-creation-time**: Get the session creation time in ISO-8601 format.
- **get-session-creation-time-millis**: Get the session creation time in millseconds since the UNIX Epoch.
- **get-session-last-accessed-time**: Get the session last accessed time in ISO-8601 format.
- **get-session-last-accessed-time-millis**: Get the session last accessed time in milliseconds since the UNIX Epoch.
- **list-session-attribute-names**: List the session attribute names.
- **list-session-attributes**: List all attributes in a session.

- **list-sessions**: List all active sessions.

2.7. IO

New Worker Attribute

In previous releases of JBoss EAP, the core threads size was always equal to the max threads size. This meant that threads would never die, even if the **task-keepalive** attribute was set. In this release, the number of threads for the core thread pool can be configured separately using the **task-core-threads** attribute, allowing the keepalive setting to work as expected.

For more information, see Configuring a Worker and IO Subsystem Attributes in the *Configuration Guide* for JBoss EAP.

2.8. LOGGING

Socket Log Handlers

You can now configure a socket log handler to send log messages over a TCP or UDP socket to a remote logging server.

For more information, see Configure a Socket Log Handler in the *Configuration Guide*.

JSON and XML Formatters

You can use the JSON and XML log formatters to format log messages in JSON and XML.

For more information, see Log Formatters in the *Configuration Guide*.

2.9. TRANSACTIONS

New maximum-timeout Transaction Manager Attribute

Previously, when users set a transaction timeout of **0**, which implies an unlimited timeout, the transaction manager used **Integer.MAX_VALUE** as the actual value for the transaction timeout. Because the maximum integer value could exhibit problems, the transaction timeout value is now capped at a smaller value.

A new configurable attribute, **maximum-timeout** has been added to the **transactions** subsystem with a default value of **31536000** seconds (365 days). If a transaction is configured with an unlimited timeout, the transaction manager now uses the value of **maximum-timeout** instead, and a **WARN** message notifying this behavior is logged.

2.10. DATASOURCES

Retrieve Datasource Class Properties for a JDBC Driver

The **datasource-class-info** runtime attribute provides the list of datasource connection properties that can be set for a JDBC driver's datasource class. When using the management console to add or edit an XA datasource, or edit a non-XA datasource, the properties field provides this list of properties as suggestions.

NOTE

The JDBC driver must have been created with the **driver-datasource-class-name** or **driver-xa-datasource-class-name** set for the properties to be shown. In a managed domain, the profile containing the JDBC driver must have a running server for the properties to be shown.

For more information, see the Datasource Attributes table in the *Configuration Guide*.

2.11. EJB

EJB and JNDI over HTTP/HTTPS with HTTP Load Balancer

Performing EJB and JNDI invocations using the HTTP protocol, so that requests are mapped directly to HTTP requests, is now fully supported in JBoss EAP 7.2. In addition, you can invoke EJBs over an HTTP load balancer. For more information, see EJB Invocation Over HTTP in the *Developing EJB Applications*.

Returning Context Data to EJB Clients

An **EJBClientInterceptor** can request specific data from the server side invocation context by calling **org.jboss.ejb.client.EJBClientInvocationContext#addReturnedContextDataKey(*String key*)**. If the requested data is present under the provided key in the context data map, it is sent to the client.

2.12. JSF

Disallowing DOCTYPE Declarations in JSF Deployments

You can use the management CLI to disallow **DOCTYPE** declarations in JSF deployments.

For more information, see Disallowing DOCTYPE Declarations in the *Configuration Guide*.

2.13. HIBERNATE

Upgraded from Hibernate ORM 5.1 to Hibernate ORM 5.3

JBoss EAP 7.2 now includes Hibernate ORM 5.3. Hibernate ORM 5.3 includes changes that were made for Hibernate ORM 5.2, which was built using the Java 8 JDK and required the Java 8 JRE at runtime. Hibernate ORM 5.3 also adds support for the JPA 2.2 specification. It contains changes to comply with this specification, along with other improvements.

For more information about the features introduced in Hibernate ORM 5.2 and 5.3, along with what you need to know to migrate your applications from Hibernate ORM 5.1 to Hibernate ORM 5.3, see Migrating from Hibernate ORM 5.1 to Hibernate ORM 5.3 in the *Migration Guide* for JBoss EAP.

Upgraded from Hibernate Validator 5.3.x to Hibernate Validator 6.0.x

JBoss EAP 7.2 includes Hibernate Validator 6.0.x, which is the reference implementation for JSR 380: Bean Validation 2.0.

For more information, see About Bean Validation in the *Development Guide* for JBoss EAP.

2.14. CLUSTERING

Enhanced Execute Methods Use CompletableFuture

In this release, **CommandDispatcher** asynchronous methods were enhanced to take advantage of the new Java EE 8 **CompletableFuture** interface. This allows consumers of **CommandDispatcher** to implement non-blocking handling of dispatched commands.

Deprecated Method	Replacement Method
executeOnNode	executeOnMember
executeOnCluster	executeOnGroup

Deprecated Method	Replacement Method
submitOnNode	executeOnMember
submitOnCluster	executeOnGroup

For more information, see Public API for Clustering Services in the *Development Guide*.

2.15. INFINISPAN

HotRod Client Injection

You can inject a HotRod client to connect to a remote JDG cluster using the **@Resource** JNDI injection.

For more information, see Externalize HTTP Sessions to JBoss Data Grid in the *Configuration Guide*.

Non-blocking Initial State Transfer

Caches can now be made immediately available instead of waiting for state transfer to complete. This is accomplished by setting the **timeout** attribute of the cache to **0**, allowing the cache to receive its state through background operations.

For more information, see State Transfer in the *Configuration Guide*.

Scattered Cache Mode

The **infinispan** subsystem now supports scattered cache mode. Scattered mode is similar to distributed mode in that it uses a consistent hash algorithm to determine ownership. However, ownership is limited to two members, and the originator, or node receiving the request for a given session, always assumes ownership for coordinating locking and cache entry updates. The cache write algorithm used in scattered mode guarantees that a write operation results in only a single RPC call. This can potentially reduce contention and improve performance following a cluster topology change.

For more information, see Clustering Modes in the *Configuration Guide*.

Externalize HTTP Sessions Using the Remote Cache Store

A new method of externalizing HTTP sessions to JBoss Data Grid is included in this release. This method utilizes a remote cache container in the **infinispan** subsystem of JBoss EAP that has a client SSL context defined for security.

You can configure remote cache containers from the management CLI and the management console.

For more information, see Externalize HTTP Sessions to JBoss Data Grid in the *Configuration Guide*.

2.16. WEB SERVICES

Java API for JSON Binding

RESTEasy supports both JSON-B and JSON-P. In accordance with the specification, entity providers for JSON-B take precedence over the ones for JSON-P for all types of entities except **JsonValue** and its sub-types.

The **JsonBindingProvider** property from **resteasy-json-binding-provider** module provides support for JSON-B. To satisfy JAX-RS 2.1 requirements, the **JsonBindingProvider** provider takes precedence over the other providers for dealing with JSON payloads, in particular the Jackson payload. In order to

retain backward compatibility, you can set the **resteasy.preferJacksonOverJsonB** context property to **true** and disable the **JsonBindingProvider** configuration for the current deployment.

For details, see the Java API for JSON Binding section in *Developing Web Services Applications* for JBoss EAP.

Asynchronous HTTP Request Processing

The default asynchronous engine implementation class for RESTEasy is **ApacheHttpAsyncClient4Engine**. You can set the asynchronous engine as the active engine by calling the **useAsyncHttpEngine** method in the **ResteasyClientBuilder** class.

For details, see the Asynchronous NIO Request Processing section in *Developing Web Services Applications* for JBoss EAP.

Custom RESTEasy Annotations

With the addition of parameter names in the bytecode, you are no longer required to specify the parameter names in the following annotations: **@PathParam**, **@QueryParam**, **@FormParam**, **@CookieParam**, **@HeaderParam** and **@MatrixParam**. To do so, you must switch to the new annotations with the same name, in a different package, which have an optional value parameter.

For details, see the Custom RESTEasy Annotations section in *Developing Web Services Applications* for JBoss EAP.

Extending the ParamConverter Functionality

In the JAX-RS semantics, a **ParamConverter** converts a single string that represents an individual object. RESTEasy extends the semantics to allow a **ParamConverter** to parse the string representation of multiple objects and generate a **List<T>**, **Set<T>**, **SortedSet<T>**, array, or any other multi-valued data structure.

For details, see the Extending the Functionality of the ParamConverter section in *Developing Web Services Applications* for JBoss EAP.

Resource Method Algorithm Switch

A bug discovered in the resource method matching algorithm used in RESTEasy 3.0.x versions prior to 3.0.25.Final caused RESTEasy to return too many resource methods when responding to requests. For more information, see JAX-RS and RESTEasy Application Changes in the *Migration Guide*.

RESTEasy Service Provider Interface

JBoss EAP now provides a RESTEasy service provider interface (SPI) to modify resource class metadata, which is created using **ResourceBuilder**. Implementations of the **ResourceClassProcessor** interface allows customizing the metadata generation.

For more information about the RESTEasy SPI, see the RESTEasy SPI to Modify Resource Metadata section in *Developing Web Services Applications* for JBoss EAP.

JAX-RS Client Support for HTTP Redirects

JAX-RS **ClientHttpEngine** implementations based on the Apache **HttpClient** support HTTP redirection. For more information, see HTTP Redirect in *Developing Web Services Applications* .

2.17. MESSAGING

IBM MQ Resource Adapter

This release of JBoss EAP was tested with the the following configurations.

- The IBM MQ 8.0.0.10 resource adapter was tested against the IBM MQ 8.0.0.x broker. Versions 8.0.0.0 through 8.0.0.9 of the IBM MQ resource adapter are not supported.

- The IBM MQ 9.0.0.4 resource adapter was tested against the IBM MQ 9.0.0.x broker. Versions 9.0.0.0 through 9.0.0.3 of the IBM MQ resource adapter are not supported.

For more information about the IBM MQ resource adapters, see Deploying the IBM MQ Resource Adapter in *Configuring Messaging* for JBoss EAP.

Messaging Journal Persistence Using a JDBC Database

In addition to the currently supported Oracle 12c database, this release of JBoss EAP adds support for the IBM DB2 Enterprise database when using JDBC to persist messages.

Support for HA Topology for Messaging JDBC Persistence Store

This release of JBoss EAP supports HA topology for messaging JDBC persistence store. For details, see Configuring HA for Messaging JDBC Persistence Store in *Configuring Messaging* for JBoss EAP.

Simplifying Connection to Remote Red Hat AMQ 7 Messaging Broker

Connection to Remote Red Hat AMQ 7 messaging broker no longer requires the presence of JBoss EAP's embedded messaging broker. You can define resources required for connection to remote Red Hat AMQ broker directly in the **messaging-activemq** subsystem.

Connect to Red Hat AMQ Using the Integrated Artemis Resource Adapter

You can configure the integrated Artemis resource adapter to connect to a remote installation of Red Hat AMQ 7, which then becomes the JMS provider for your JBoss EAP 7.2 applications. This allows JBoss EAP to be a client for the remote Red Hat AMQ 7 server.

For more information, see Configuring the Artemis Resource Adapter to Connect to Red Hat JBoss AMQ 7 in *Configuring Messaging* for JBoss EAP.

For more information about configuring the **journal-file-open-timeout** attribute, see Configuring Message Journal Attributes in the *Configuring Messaging* book for JBoss EAP.

Change in Artemis Logging Codes

Artemis logging codes for Artemis core protocol have changed, whereas the Advanced Message Queuing Protocol (AMQP) codes remain the same. This creates a problem if you are monitoring issues based on these codes.

The logging codes changed because the codes were duplicated between AMQP and the Artemis core protocol.

2.18. OPENSHIFT

KUBE_PING Integrated Natively In JBoss EAP

Previously, the **KUBE_PING** JGroups discovery protocol was implemented only in the JBoss EAP OpenShift image. **KUBE_PING** is now implemented natively in JBoss EAP, so users creating their own custom container images are now able to natively use **KUBE_PING** for clustered applications. For more information on using **KUBE_PING**, see the Clustering reference in *Getting Started with JBoss EAP for OpenShift Container Platform*.

2.19. MODULES

Predefined Modules

A set of predefined modules, **org.jboss.modules**, which includes all of the JBoss Modules API, is supported in JBoss EAP 7.2 when you use the default module loader. This special module is always available and is provided by JBoss Modules. The standard Java Platform Module System (JPMS) modules, which are provided in Java 9 and later, are also available by their standard names. When using JDK 8, the JDK 9 modules are emulated by JBoss Modules.

For more information, see Predefined Modules in the *Configuration Guide*.

2.20. QUICKSTARTS AND BOMS

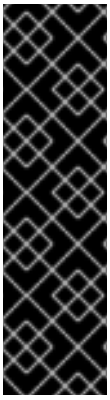
JBoss EAP BOMs Available for Application Development

The artifact IDs for JBoss EAP Maven BOM files have changed because of the update to Java EE 8. The following table lists the Maven BOMs that are available for application development in this release.

BOM Artifact ID	Use Case
jboss-eap-javaee8	Supported JBoss EAP Java EE 8 APIs plus additional JBoss EAP API JARs.
jboss-eap-javaee8-with-spring4	**jboss-eap-javaee8** plus recommended Spring 4 versions.
jboss-eap-javaee8-with-tools	**jboss-eap-javaee8** plus development tools such as Arquillian.

For more information about the BOMs available for application development, see Manage Project Dependencies in the *Development Guide*.

CHAPTER 3. TECHNOLOGY PREVIEW



IMPORTANT

The following configurations and features are provided as Technology Preview only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs), might not be functionally complete, and Red Hat does not recommend to use them for production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

See Technology Preview Features Support Scope on the Red Hat Customer Portal for information about the support scope for Technology Preview features.

New Agroal Datasources Subsystem

JBoss EAP 7.2 introduces the **datasources-agroal** subsystem, which is a lightweight datasource connection pool implementation that performs exceptionally well in situations with high contention on the pool. This can be used as an alternative to the current JCA-based **datasources** subsystem.

This subsystem is not enabled in the default JBoss EAP configuration. For information on enabling and using Agroal datasources, see Datasource Management With Agroal in the *Configuration Guide*.

MicroProfile REST Client

JBoss EAP now supports the MicroProfile REST client that builds on JAX-RS 2.0 client APIs to provide a type-safe approach to invoke RESTful services over HTTP. With the MicroProfile REST client, you can write client applications with executable code. The MicroProfile TypeSafe REST clients, which are defined as Java interfaces, enable programmatic and declarative registration of providers.

For more information about the MicroProfile REST client, see the MicroProfile Rest Client section in *Developing Web Services Applications* for JBoss EAP.

Extending RESTEasy Support for Asynchronous Request Processing and Reactive Return Types

JBoss EAP now extends the RESTEasy support for asynchronous request processing and reactive return types. This includes support for pluggable reactive types, and extensions for additional reactive classes. A new type of invoker named **RxInvoker**, and a default implementation of this type named **CompletionStageRxInvoker** are also supported. JBoss EAP also provides the ability to convert a filter into an asynchronous filter. Proxies, the RESTEasy extension that supports an intuitive programming, is extended to include both **CompletionStage** and the RxJava2 types **Single**, **Observable**, and **Flowable**.

For more information, see the Extending RESTEasy Support for Asynchronous Request Processing and Reactive Return Types section in *Developing Web Services Applications* for JBoss EAP.

Eclipse MicroProfile Config

JBoss EAP 7.2 implements the SmallRye Config component, which provides support for Eclipse MicroProfile Config using the **microprofile-config-smallrye** subsystem. This allows applications and microservices to be configured to run in multiple environments without a need for modification or repackaging.

For more information, see Using Eclipse MicroProfile Config to Manage Configuration in the *Configuration Guide*.

Eclipse MicroProfile OpenTracing

JBoss EAP 7.2 implements the SmallRye OpenTracing component, which provides support for the Eclipse MicroProfile OpenTracing specification. This allows requests to be traced as they go through

applications and services deployed to the JBoss EAP server, and extends observability of the request's lifecycle.

For more information about the **microprofile-opentracing-smallrye** subsystem, see Tracing Requests with the MicroProfile OpenTracing SmallRye Subsystem in the *Configuration Guide*. For information about how to customize tracing for CDI beans and JAX-RS endpoints, see Using Eclipse MicroProfile OpenTracing to Trace Requests in the *Development Guide*.

Eclipse MicroProfile Health

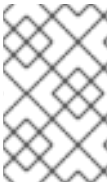
JBoss EAP 7.2 includes the SmallRye Health component, which provides Eclipse MicroProfile Health functionality. This feature allows you to check on the health of a remote node and determine if the JBoss EAP instance is responding as expected. The **microprofile-health-smallrye** subsystem is included in the default JBoss EAP 7.2 configuration.

For more information about the **microprofile-health-smallrye** subsystem, see Monitor Server Health Using the MicroProfile Health Check in the *Configuration Guide*. See Implement a Custom Health Check in the *Development Guide* for information about how to implement a custom health check.

CHAPTER 4. UNSUPPORTED AND DEPRECATED FUNCTIONALITY

4.1. UNSUPPORTED FEATURES

Support for some technologies are removed due to the high maintenance cost, low community interest, and better alternative solutions. The following features are not supported in JBoss EAP 7.2.



NOTE

The unsupported features listed in the Unsupported Features section of the *7.1.0 Release Notes* also apply to JBoss EAP 7.2, unless they are mentioned in the New Features and Enhancements section of this document.

Platforms and Features

This release has removed support for the following platforms and features:

Operating Systems and Related Web Servers

- Windows Server 2008 and associated IIS web server
- Solaris 10 and associated web servers
- HP-UX
- Red Hat Enterprise Linux 6 32-bit

Java Virtual Machine

- HP-UX

Databases and Database Connectors

- IBM DB2 e9.7 and e10.5
- MySQL 5.5
- Microsoft SQL Server 2012 SP3
- Microsoft SQL Server 2014 SP2
- PostgreSQL 9.3, 9.4, and 9.6
- EnterpriseDB 9.3, 9.4, and 9.6
- Sybase ASE 15.7
- Oracle 11g R2 and 11g R2 RAC

JMS Providers and Adapters

- IBM WebSphere MQ 7.5

LDAP Servers

- Red Hat Directory Server 9.1
- Microsoft Active Directory 2008

Tested Frameworks

- JQuery (all versions)
- AngularJS (all versions)

Messaging (ActiveMQ Artemis)

Configuring a discovery group or a broadcast group using JGroups is not supported in a cluster consisting of different versions of JBoss EAP. For more information, see Clusters Overview in *Configuring Messaging* and Upgrading a Cluster in the *Patching and Upgrading Guide*.

RPM Current Repository

The JBoss EAP **current** repository is not available for this release. If you are subscribed to the JBoss EAP **current** repository, you must change your subscription to a JBoss EAP **minor** repository. For more information, see Changing Repositories in the *Installation Guide*. Information about Patching an RPM Installation and Upgrading an RPM Installation is available in the *Patching and Upgrading Guide*.

Quickstarts

The following quickstarts, which were available in JBoss EAP 7.1, are no longer available in this release.

- cdi-alternative
- cdi-decorator
- cdi-injection
- cdi-interceptors
- cdi-portable-extension
- cdi-stereotyp
- cdi-veto
- ejb-security-interceptors
- forge-from-scratch
- h2-console
- kitchensink-html5-mobile
- kitchensink-ml-ear
- log4j
- picketlink-sts
- shrinkwrap-resolver
- spring-kitchensink-asyncrequestmapping

- spring-kitchensink-controlleradvice
- spring-kitchensink-matrixvariables
- spring-petclinic
- tasks
- xml-dom4j

Internal Datasources and Drivers for OpenShift JDK 11 image

The following internal datasources and drivers are no longer provided with the JBoss EAP for OpenShift JDK 11 image:

- MySQL
- PostgreSQL
- MongoDB

It is recommended that you use JDBC drivers obtained from your database vendor for your JBoss EAP applications.

For more information about installing drivers, see the Modules, Drivers, and Generic Deployments section in Getting Started with JBoss EAP for OpenShift Container Platform .

For more information on configuring JDBC drivers with JBoss EAP, see the JDBC drivers section in the JBoss EAP Configuration Guide.

4.2. DEPRECATED FEATURES

Some features have been deprecated with this release. This means that no enhancements will be made to these features, and they may be removed in the future, usually the next major release.

Red Hat will continue providing full support and bug fixes under our standard support terms and conditions. For more information about the Red Hat support policy, see the Red Hat JBoss Middleware Product Update and Support Policy located on the Red Hat Customer Portal.

For details of which features have been deprecated, see the JBoss Enterprise Application Platform Component Details located on the Red Hat Customer Portal.

IO Subsystem

- IO buffer pools are deprecated in this release. They are replaced by Undertow byte buffer pools.

Cache Stores

- The **remote** cache store has been deprecated in favor of using the **hotrod** cache store.

Red Hat JBoss Operations Network

The use of Red Hat JBoss Operations Network for managing JBoss EAP is deprecated with this release.

Platforms and Features

Support for the following platforms and features is deprecated:

Operating Systems and Related Web Servers

- Windows Server 2012 R2 and associated IIS web server
- Solaris 11 on SPARC and associated web servers
- Solaris 11 on x86_64 and associated web servers

JMS Providers and Adapters

- TIBCO EMS

LDAP Servers

- Microsoft Active Directory 2012 R2

Tested Frameworks

- Spring (Core), Spring Security, Spring Web Flow, Spring Web Services

CHAPTER 5. RESOLVED ISSUES

See Resolved Issues for JBoss EAP 7.2 GA to view the list of issues that have been resolved for this release.

CHAPTER 6. FIXED CVES

JBoss EAP 7.2 includes fixes for the following security-related issues:

- CVE-2017-7503: **xml frameworks**: JBoss EAP 7.0.5 implementation of **javax.xml.transform.TransformerFactory** is vulnerable to XXE
- CVE-2018-10237: **guava**: Unbounded memory allocation in **AtomicDoubleArray** and **CompoundOrdering** classes allow remote attackers to cause a denial of service
- CVE-2018-1067: **undertow**: HTTP header injection using CRLF with UTF-8 encoding
- CVE-2018-10862: **wildfly-core**: Path traversal can allow the extraction of **.war** archives to write arbitrary files
- CVE-2017-12174: **artemis/hornetq**: Memory exhaustion via UDP and JGroups discovery
- CVE-2017-12629: **Solr**: Code execution via entity expansion
- CVE-2017-15089: **infinispan**: Unsafe deserialization of malicious object injected into data cache
- CVE-2017-12196: **undertow**: Client can use bogus uri in Digest authentication
- CVE-2018-8088: **slf4j**: Deserialisation vulnerability in EventData constructor can allow for arbitrary code execution
- CVE-2018-1047: **undertow**: Path traversal in ServletResourceManager class
- CVE-2018-8039: **apache-cxf**: TLS hostname verification does not work correctly with **com.sun.net.ssl.***

CHAPTER 7. KNOWN ISSUES

See Known Issues for JBoss EAP 7.2 GA to view the list of known issues for this release.

Additionally, be aware of the following:

- If you try to start an embedded server from a management CLI instance that was started using the **jboss-cli-client.jar** file on JDK 11, you will get an error: **WFLYEMB0014: Cannot load module**. To avoid this error, you must add **--add-modules java.se** when starting the management CLI:

```
$ java --add-modules java.se -jar jboss-cli-client.jar
```

- By default, JBoss EAP allows duplicate XA resources to return **XAER_PROTO** for duplicate **xa_end** calls, as opposed to the expected JTA 1.2 behavior of allowing each resource to perform an **xa_end** call. This is because JBoss EAP ships with Artemis, which does not currently support the JTA 1.2 behavior for duplicate **xa_end** calls, as described in JBEAP-12671. You can enable this strict behavior for JBoss EAP to be fully JTA 1.2 compliant by passing in the flag **-DJTAEnvironmentBean.strictJTA12DuplicateXAENDPROTOErr=true**. However, this will result in errors when using Artemis, and transactions that include duplicate Artemis resources will not be able to commit.

- If a client application directly depends on Artemis client JARs, for example, **artemis-jms-client**, **artemis-commons**, **artemis-core-client**, or **artemis-selector**, then you must add a dependency in your **pom.xml** file for **wildfly-client-properties**:

```
<dependency>
  <groupId>org.jboss.eap</groupId>
  <artifactId>wildfly-client-properties</artifactId>
</dependency>
```

  This is to avoid a **JMSRuntimeException** when calling **message.getJMSReplyTo()** from an older JBoss EAP 7 client as described in JBEAP-15889.

- If you install the JBoss EAP 7.2 RPM packages on RHEL 7 and it only includes the default RHEL JDK 11 package, due to the dependencies currently defined in the packages, the RPM will also install JDK 1.8. This will result in JDK 1.8 being set as the default JDK. For this reason, you must use **alternatives** to configure the system to use JDK 11.

```
alternatives --config java
```

  Enter the number that corresponds to JDK 11.

- Prior to JBoss EAP 7.2, services defined in global modules were accessible from external dependencies, even if they were not configured to be exposed externally. In JBoss EAP 7.2, this behavior was corrected. Since JBoss EAP 7.2, if you want to make a service in a global module available externally, you must correctly configure them. Services that are not explicitly configured are not exposed. For details about how to configure a service to be exposed externally, see Define Global Modules in the Configuration Guide for JBoss EAP.

Revised on 2019-09-26 12:52:54 UTC