Red Hat JBoss Enterprise Application Platform 7.1

7.1.0 Release Notes

For Use with Red Hat JBoss Enterprise Application Platform 7.1

Last Updated: 2018-06-27

Red Hat JBoss Enterprise Application Platform 7.1 7.1.0 Release Notes

For Use with Red Hat JBoss Enterprise Application Platform 7.1

Legal Notice

Abstract

These release notes contain important information related to Red Hat JBoss Enterprise Application Platform 7.1.

Table of Contents

CHAPTER 1. ABOUT RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 7.1

Red Hat JBoss Enterprise Application Platform 7.1 (JBoss EAP) is a middleware platform built on open standards and compliant with the Java Enterprise Edition 7 specification.

JBoss EAP includes a modular structure that allows service enabling only when required, improving startup speed.

The management console and management command-line interface (CLI) make editing XML configuration files unnecessary and add the ability to script and automate tasks.

JBoss EAP provides two operating modes for JBoss EAP instances: standalone server or managed domain. The standalone server operating mode represents running JBoss EAP as a single server instance. The managed domain operating mode allows for the management of multiple JBoss EAP instances from a single control point.

In addition, JBoss EAP includes APIs and development frameworks for quickly developing secure and scalable Java EE applications.

CHAPTER 2. SUPPORTED CONFIGURATIONS

The following configurations are newly supported for JBoss EAP 7.1.

- Operating systems
  - Windows Server 2016 on x86_64 architecture
    - This includes using JBoss EAP in Microsoft Azure on a Windows Server 2016 virtual machine.
- Databases
  The following databases have been certified and are now fully supported:
  - SQL Server 2016
  - Sybase 16.0
  - MariaDB Galera Cluster 10.1
- External JMS providers
  - Red Hat JBoss AMQ 7.0
  - IBM WebSphere MQ 8
- LDAP services
  - Red Hat Directory Server 10.1
  - Microsoft Active Directory 2016
- Native connectors
  - Microsoft IIS 10
- Red Hat JBoss Developer Studio
  - JBoss EAP 7.1 is certified for use with Red Hat JBoss Developer Studio 11.

See the Red Hat JBoss Enterprise Application Platform (EAP) 7 Supported Configurations page for full supported configuration details for JBoss EAP 7.1.

CHAPTER 3. NEW FEATURES AND ENHANCEMENTS

3.1. SECURITY AND ELYTRON

Elytron and the elytron Subsystem

The **elytron** subsystem, which is based on the WildFly Elytron project, is new in JBoss EAP 7.1. Elytron is a security framework used to unify security across the entire application server. The **elytron** subsystem provides a single point of configuration for securing both applications and the management interfaces. It provides a set of APIs and SPIs for creating custom implementations of functionality and integration. To learn more about the various Elytron components, see Core Concepts and Components section of the *Security Architecture* guide.

The legacy **security** subsystem and legacy core management authentication are still present in JBoss EAP 7.1 and are used by default. You can find information on configuring the **elytron** subsystem in the Elytron Subsystem section of *How to Configure Server Security.*

Important features of the **elytron** subsystem include:

- Stronger authentication mechanisms for HTTP and SASL authentication.
- An improved architecture that allows security identities to be propagated across security domains and transparently transformed to make them ready to use for authorization. Transformation takes place using configurable role decoders, role mappers, and permission mappers.
- A centralized point for SSL/TLS configuration, including cipher suites and protocols.
- SSL/TLS optimizations such as eager secure identity construction and closely tying authorization to establishing an SSL/TLS connection. This enables permission checks to happen before the first request is received. Eager secure identity construction eliminates the need for a secure identity to be constructed on a per-request basis.
- A secure credential store that replaces the legacy password vault implementation. The secure credential store can store multiple other encrypted credential types in addition to encrypted strings. You can find more information on credential stores in the Credential Store section of *How to Configure Server Security.* With the exception of the **elytron** subsystem, new and existing legacy password vaults can still be used with other subsystems.

Credential Stores

You can configure credential stores in the **elytron** subsystem for JBoss EAP 7.1. A credential store allows for secure storage and usage of credentials, and has many benefits compared to using a legacy password vault. Credentials stored in a credential store can be securely referenced by other JBoss EAP subsystems. This prevents credentials, such as passwords, from being stored in plain text. For more information, see Credential Store in *How to Configure Server Security.*

Mapping Identity for Authenticated Management Users

When using the **elytron** subsystem to secure the management interfaces, you can provide a security domain to the management interfaces for identity mapping of authenticated users. This allows authenticated users to appear with the appropriate identity when logged into the management interfaces. For more information, see Mapping Identity for Authenticated Management Users in *How to Configure Server Security.*

Automatic Self-signed Certificate Creation for Applications

JBoss EAP 7.1 provides automatic generation of a self-signed certificate for development purposes for legacy security realms. For more information, see Automatic Self-signed Certificate Creation for Applications in *How to Configure Server Security*.

Caching for Security Realms

Elytron provides a **`caching-realm`** which allows you to cache the results of a credential lookup from a security realm. For example, you could use this to configure a cache for credentials coming from LDAP or a database to increase performance for frequently queried users. For more information, see Set Up Caching for Security Realms in *How to Configure Identity Management*.

Container-Managed Single Sign-on

You can configure JBoss EAP 7.1 to use container-managed single sign-on for applications using the Elytron **`FORM`** authentication method. This allows users to authenticate once and access other resources secured by the **`FORM`** authentication method without having to reauthenticate. For more information, see Configure Applications to use Container-managed Single Sign-on in *How to Configure Identity Management*.

Propagating Security Identities for Remote Calls

JBoss EAP 7.1 introduces the ability to easily configure the server and your applications to propagate a security identity from a client to the server for remoting calls. You can also configure server components to run within the security identity of a given user.

For more information, see Propagating Security Identities for Remote Calls in *How to Configure Server Security* for JBoss EAP.

WildFly Elytron Tool

JBoss EAP 7.1 includes the WildFly Elytron Tool, which allows you to create and modify credential stores without needing a running JBoss EAP server. It can also be used to convert password vaults to credential stores by using the **`vault`** option.

See Create and Modify Credential Stores Offline with the WildFly Elytron Tool in *How to Configure Server Security* for information on how to use the WildFly Elytron Tool.

Script to Enable Elytron in Applicable Subsystems and Management Interfaces

A script is provided to enable the Elytron framework in applicable subsystems and management interfaces. This script, **`enable-elytron.cli`** is available in the ***`EAP_HOME`*`/docs/examples/`** directory. The use of this script is optional; Elytron can also be enabled on individual subsystems as needed. For more information, see How Red Hat JBoss Enterprise Application Platform 7.1 Handles Security out of the Box in the *Security Architecture* guide.

Configuring the Elytron Subsystem Using the Management Console

You can configure the **`elytron`** subsystem using the management console by navigating to **Configuration** → **Subsystems** → **Security - Elytron**. For more information, see Elytron Subsystem in *How to Configure Server Security*.

Elytron Integration With the JBoss EAP Subsystems

In JBoss EAP 7.1, you can use Elytron to secure various aspects of the following JBoss EAP subsystems:

`batch-jberet`

You can configure the **`batch-jberet`** subsystem to run batch jobs using an Elytron security domain. For more information, see Configure Security for Batch Jobs in *Configuration Guide*.

`datasources`

You can use a credential store or an Elytron security domain to provide authentication information in a datasource definition. For more information, see Datasource Security in *Configuration Guide*.

`ejb3`

You can create mappings for Elytron security domains in the **`ejb3`** subsystem to be referenced by deployments. For more information, see Elytron Integration with the EJB Subsystem in *Developing EJB Applications*.

`iiop-openjdk`

You can configure the **`iiop-openjdk`** subsystem to use SSL/TLS to secure communication between clients and servers. For more information, see Configure IIOP to Use SSL/TLS with the Elytron Subsystem in the *Configuration Guide*.

`jca`

You can use the **`elytron-enabled`** attribute to enable Elytron security for a work manager. For more information, see Configuring the JCA Subsystem in the *Configuration Guide*.

`jgroups`

You can configure the **`SYM_ENCRYPT`** and **`ASYM_ENCRYPT`** protocols to reference keystores or credential stores defined in the **`elytron`** subsystem. The **`AUTH`** protocol can also be configured to reference elytron-managed credential stores and keystores. For more information, see Securing a Cluster in the *Configuration Guide*.

`mail`

You can use a credential store to provide passwords for the **`mail`** subsystem. For more information, see Use a Credential Store for Passwords in the *Configuration Guide*.

`messaging-activemq`

You can use Elytron security to secure the **`messaging-activemq`** subsystem. For more information, see the Using the Elytron Subsystem section of *Configuring Messaging*.

`modcluster`

You can use an Elytron client **`ssl-context`** to communicate with a load balancer using SSL/TLS. For more information, see Elytron Integration with the ModCluster Subsystem in *How to Configure Server Security*.

`remoting`

You can configure inbound and outbound connections in the **`remoting`** subsystem to reference authentication contexts, SASL authentication factories, and SSL contexts defined in the **`elytron`** subsystem. For more information, see Elytron Integration with the Remoting Subsystem in *How to Configure Server Security*.

`resource-adapters`

You can secure connections to the resource adapter using Elytron. You can enable security inflow to establish security credentials when submitting work to be executed by the work manager. For more information, see Configure Resource Adapters to Use the Elytron Subsystem in the *Configuration Guide*.

`undertow`

You can use the elytron subsystem to configure both SSL/TLS and application authentication. For more information, see Using SSL/TLS in *How to Configure Server Security* and Configure Web Applications to Use Elytron or Legacy Security for Authentication in *How to Configure Identity Management*.

3.2. SERVER MANAGEMENT

Starting Servers in a Suspended State

During the startup process, JBoss EAP 7.1 servers are left in a suspended state until all services are started. While in this state, the server does not accept any requests. Once all required services have started, the server is placed into a normal running state to start accepting requests.

It is also possible to start servers in a suspended state and keep them suspended until the **resume** operation is invoked. To start the server in a suspended state, set the **start-mode** argument to **suspend** for the appropriate operation.

- For a standalone server, pass the **--start-mode=suspend** argument to the **standalone.sh** script:

  Example: Start a Standalone Server in a Suspended State

  ```
  $ EAP_HOME/bin/standalone.sh --start-mode=suspend
  ```

- In a managed domain, pass the **start-mode=suspend** argument to the **start** management CLI operation:

  Example: Start a Managed Domain Server in a Suspended State

  ```
  /host=HOST_NAME/server-config=SERVER_NAME:start(start-mode=suspend)
  ```

Monitoring Server Lifecycle Events Using the Core Management Subsystem

In JBoss EAP 7.1, you can register a listener to the JBoss EAP **core-management** subsystem to monitor for server lifecycle events. For more information, see Monitor Server Lifecycle Events Using the Core Management Subsystem in the *Configuration Guide*.

Monitoring Server Lifecycle Events Using JMX Notifications

In JBoss EAP 7.1, you can register a JMX notification listener to monitor for server lifecycle events. For more information, see Monitor Server Lifecycle Events Using JMX Notifications in the *Configuration Guide*.

Tracking and Viewing Configuration Changes from the Management CLI

In a managed domain, configuration changes are tracked at the host level for host and server-related modifications. Enabling configuration changes for a host controller enables it for all of its managed servers. Configuring tracking configuration changes has been moved to the new **core-management** subsystem. For more information, see View Configuration Changes in the *Configuration Guide*.

Monitoring Worker Statistics

You can view a worker's runtime statistics using the management CLI. This exposes worker statistics such as connection count, thread count, and queue size.

The following command displays runtime statistics for the default worker:

```
/subsystem=io/worker=default:read-resource(include-
runtime=true,recursive=true)
```

For more information, see Configuring Workers in the *Performance Tuning Guide*.

Improved Resource Monitoring for Slave Host Controllers

In JBoss EAP 7.1, host controllers that are configured as slaves can ignore resources in the domain-wide configuration that are not required. Resources can be irrelevant if they are not associated with the servers managed by the slave host controllers.

You can ignore unused configuration by setting the **ignore-unused-configuration** attribute to **true** in the JBoss EAP 7.0 host controller's connection configuration to the remote domain controller. By default, the **ignore-unused-configuration** attribute is not defined.

For more information and an example configuration, see Configure a JBoss EAP 7.1 Domain Controller to Administer JBoss EAP 7.0 Instances in the *Configuration Guide*.

You can also use the **--backup** command line flag along with **ignore-unused-configuration** set to **true**, which allows a slave host controller to start using a backup copy of the domain configuration if the domain controller is not available. The slave host controller does not require the full **domain.xml** to do this.

Host Controllers Started Using a Cached Configuration Automatically Reconnect to the Domain Controller

In JBoss EAP 7.1, a host controller that has been started using a cached configuration because the domain controller was unreachable will automatically reconnect once the domain controller becomes available.

Setting the Server Locale

You can use the **org.jboss.logging.locale** property to override the locale for messages logged using JBoss Logging, including any messages from JBoss EAP and its owned dependencies.

For more information, see Set the Server Locale Using the org.jboss.logging.locale Property in the *Configuration Guide*.

New Attribute: parse-group-name-from-dn

In JBoss EAP 7.1, the **parse-group-name-from-dn** attribute is now available at **/core-service=management/security-realm=realm/authorization=ldap/group-search=principal-to-group**. The attribute is provided in place of the **org.jboss.as.domain.management.security.parseGroupNameFromLdapDN** system property.

For more information, see Enabling the LDAP Security Realm to Parse Roles from a DN in the *Migration Guide*.

Managing JBoss EAP Using JBoss Operations Network

You can monitor JBoss EAP 7.1 servers and manage their configuration using Red Hat JBoss Operations Network.

IMPORTANT

JBoss Operations Network does not include support for configuring the new JBoss EAP 7.1 **elytron** subsystem. The monitoring support is limited to the features of the JBoss Operations Network JBoss EAP plugin that were available for JBoss EAP 6.4, with the addition of the JBoss EAP **undertow**, **iiop-openjdk**, **io**, and **messaging-activemq** subsystems.

3.3. MANAGEMENT CLI

Displaying and Saving Attachments

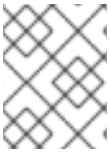
In JBoss EAP 7.1, you can use the **attachment** command to display or save the contents of an attached stream. This works for management resources that can expose contents as a stream.

Use the following management CLI command to display the contents of an attachment:

```
attachment display --operation=/subsystem=logging/log-
file=server.log:read-attribute(name=stream)
```

Use the following management CLI command to save the contents of an attachment to a file:

```
attachment save --operation=/subsystem=logging/log-file=server.log:read-
attribute(name=stream) --file=test.log
```



NOTE

If a file name is not provided, then the ***EAP_HOME*/bin/*STREAM_UUID*** is used as the file path.

See Display the Contents of an Attachment and Save the Contents of an Attachment in the *Management CLI Guide* for more information.

Attaching Files to Management Operations

In JBoss EAP 7.1, you can use the management CLI to attach a file to a management operation. You can use the **add-content** operation to add content to an existing exploded deployment or the **remove-content** operation to remove content. For example:

```
/deployment=test.war:add-content(content=[{input-stream-
index=/path/to/a.txt,target-path=a.txt}])
```

You can use the **browse-content** operation to browse the contents of a deployment.

Setting a Timeout for Commands

JBoss EAP 7.1 allows you to set the maximum time, in seconds, to wait for a management CLI command to complete. A value of **0** means no timeout. By default, there is no timeout. For example:

```
command-timeout set 30
```

Include the Prompt and Command in the Output in Non-Interactive Mode

In JBoss EAP 7.1, the **--echo-command** argument displays the prompt and command with the output for commands executed in non-interactive mode. This can be useful when resolving failures by matching the output to the command that was executed.

```
$ EAP_HOME/bin/jboss-cli.sh --connect --file=/path/to/cli_commands.txt --
echo-command
```

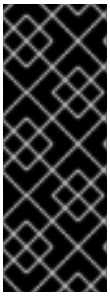
The command and its output are displayed as it executes.

```
[standalone@localhost:9990 /] :read-attribute(name=running-mode)
{
    "outcome" => "success",
    "result" => "NORMAL"
}
[standalone@localhost:9990 /] ls /deployment
helloworld.war
```

Specifying Exported Dependencies for a Custom Module

JBoss EAP 7.1 provides the **--export-dependencies** argument to specify exported dependencies for a module. For example:

```
module add --name=com.mysql --resources=/path/to/mysql-connector-java-
5.1.36-bin.jar --export-dependencies=javax.api,javax.transaction.api
```



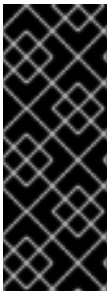
IMPORTANT

Using the **module** management CLI command to add and remove modules is provided as technology preview only. This command is not appropriate for use in a managed domain or when connecting to the management CLI remotely. Modules should be added and removed manually in a production environment. For more information, see the Create a Custom Module Manually and Remove a Custom Module Manually sections of the JBoss EAP *Configuration Guide*.

Setting an Alternative Module Directory During Module Creation

If you have defined an external JBoss EAP modules directory to use instead of the default ***EAP_HOME*/modules/** directory, you can use the **--module-root-dir** argument to specify the directory to use during module creation.

```
module add --module-root-dir=/path/to/my-external-modules/ --
name=com.mysql --resources=/path/to/mysql-connector-java-5.1.36-bin.jar --
dependencies=javax.api,javax.transaction.api
```



IMPORTANT

Using the **module** management CLI command to add and remove modules is provided as technology preview only. This command is not appropriate for use in a managed domain or when connecting to the management CLI remotely. Modules should be added and removed manually in a production environment. For more information, see the Create a Custom Module Manually and Remove a Custom Module Manually sections of the JBoss EAP *Configuration Guide*.

Starting a Management CLI Session Using the IBM JDK

The **jboss-cli** scripts set the **com.ibm.jsse2.overrideDefaultTLS** property to **true**. This setting is important if you are using the IBM JDK, to prevent authentication issues when using SSL configured by Elytron. Be sure to set this property if you are using the IBM JDK and using another method to start a CLI session, for example, programmatically using the classes available in ***EAP_HOME*/bin/client/jboss-cli-client.jar**.

3.4. MANAGEMENT CONSOLE

Application Deployment Updates

JBoss EAP 7.1 includes an updated user interface for managing application deployments. The **Deployments** tab in the management console now has the following features for deployments:

- An **Explode** drop-down option, which lets you unzip a disabled deployment.
- A **Browse Content** drop-down option, which lets you browse the files in the deployment. Navigation is not supported.
- Details about whether the application is an archive or an exploded deployment.

Transaction Monitoring Support
JBoss EAP 7.1 provides enhanced `transactions` subsystem metrics as well as metrics of JDBC and JMS transaction resources in the management console.

Viewing and Managing Messaging Prepared Transactions
You can use the management console to view, commit, or roll back prepared transactions for the `messaging-activemq` subsystem. For more information, see Manage Prepared Transactions Using the Management Console in *Configuring Messaging*.

Text Field Suggestions
As you type in some text fields in the management console, values from elsewhere in the configuration may appear as suggestions.

Adding a JMS Bridge
You can use the management console to add a JMS bridge by navigating to **Configuration** → **Subsystems** → **Messaging - ActiveMQ** → **JMS Bridge** → **View** → **Add**. Provide the required information and click **Save**.

Tracking and Viewing Configuration Changes
To enable tracking of configuration changes from the management console, navigate to the **Runtime** tab, select the standalone server or managed domain host, and select **Configuration Changes** from the drop down. Click the **Enable** button and provide a maximum history value.

The table on this page then lists each configuration change made, with the date, origin, outcome, and operation details.

Configuring Filters
You can configure Undertow filters using the management console by navigating to **Configuration** → **Subsystems** → **Web/HTTP - Undertow** → **Filters** → **View**.

Managing Batch Jobs
In JBoss EAP 7.1, you can manage batch jobs from the management console. Navigate to the **Runtime** tab, select the server, and select **Subsystems** → **Batch** → **View**. Open the **Jobs** tab and start, stop, or restart jobs as necessary.

Testing Datasource Connections
When using the **Create Datasource** wizard in the management console, you have the opportunity to test the connection before creating the datasource. On the **Test Connection** screen of the wizard, click the **Test Connection** button.

Using Datasource Templates
When creating a datasource using the management console, the **Create Datasource** wizard provides templates with default values for the supported databases. This is newly supported for JBoss EAP 7.1.

Subsystem Support
In JBoss EAP 7.1, it is now supported to configure the following subsystems using the management console:

- BeanValidation
- IO
- Jaxrs
- Jdr

- Jsf
- Jsr77
- Naming
- Pojo
- Remoting
- RequestController
- Sar
- Security - Elytron
- Singleton
- Weld

3.5. WEB SERVER

HTTP/2 Support

JBoss EAP 7.1 supports secure HTTP/2 on all supported operating systems, with the exception of HP-UX. There are two supported ways to enable HTTP/2 in JBoss EAP 7.1:

- Using JBoss EAP 7.1 internal support for ALPN, which uses the reflection API. This works out of the box, but is limited to only OpenJDK and Oracle JDK.
- Using ALPN support from the new JBoss Core Services OpenSSL, which works on all supported operating systems with the exception of HP-UX.
  - You can download JBoss Core Services OpenSSL from the JBoss Core Services OpenSSL download page.

3.6. LOGGING

Improved Reporting for Boot Errors Caused by Invalid Server Configuration Files

Prior to JBoss EAP 7.1, boot errors that occurred when parsing invalid server configuration files provided little feedback and were difficult to debug. JBoss EAP 7.1 uses XSD analysis to produce more informative error messages when encountering XML parsing errors. It now shows where the error occurred, provides feedback about the validation error, and, when possible, pulls and displays supporting documentation from the XSD to describe the issue. The enhanced validation of XML configuration does not include deployment descriptors of deployments.

Server Log Includes Patch Information

Patch-related information is now logged in the `server.log` file during startup. This information is useful while debugging issues.

3.7. DEPLOYMENTS

Managing Exploded Deployments

In JBoss EAP 7.1, you can create exploded managed deployments and manipulate their contents using deployment management operations.

For more information, see Managing Exploded Deployments in the *Configuration Guide*.

Support for Browsing the Content Repository

In JBoss EAP 7.1, you can view the contents of managed deployments using deployment management operations. For more information, see Viewing Deployment Content in the *Configuration Guide*.

Undeploying All Deployments

In JBoss EAP 7.1, you can now undeploy all deployments from the management CLI by using a wildcard (`*`). For example:

```
undeploy *
```

Redeploying All Disabled Deployments

In JBoss EAP 7.1, you can now deploy all disabled deployments from the management CLI by using a wildcard (`*`). For example:

```
deploy --name=*
```

3.8. CLASS LOADING

Using Absolute Paths for Resources in module.xml Files

In JBoss EAP 7.1, using absolute paths in the **`resource-root`** path element of the **`module.xml`** file for modules is now supported. This allows your resource libraries to be accessible without needing to move them to the ***`EAP_HOME`*`/modules/`** directory.

3.9. NAMING

Changing JNDI Bindings Dynamically

In JBoss EAP 7.1, you can use the **`rebind`** operation to update JNDI bindings dynamically without needing to reload or restart services. However, this does not work for external context bindings, as they require services to be restarted.

For more information, see the Dynamically Change JNDI Bindings section of the *Configuration Guide*.

3.10. TRANSACTIONS

Graceful Shutdown for Transactions

Once suspended, the server will not accept new requests, but in-flight transactions and requests are allowed to continue until they complete or until the timeout period expires. This also applies for web service requests associated with an XTS transaction. See Suspend and Shut Down JBoss EAP Gracefully in the *Configuration Guide* for more information.

Enhanced Transaction Monitoring

JBoss EAP 7.1 provides enhanced statistics for transaction resources in the **`datasources`**, **`transactions`**, and **`messaging-activemq`** subsystems.

See Datasource Statistics and View Transaction Statistics in the *Configuration Guide*, and Monitoring Messaging Statistics in *Configuring Messaging* for information on viewing the available statistics.

Forget Call When Deleting a Transaction

When using the **`delete`** operation on a transaction log, the **`forget`** call is now triggered so that XA resource vendor logs are cleaned correctly. For more details and how to configure the **`forget`** call behavior, see Delete a Transaction in the *Configuration Guide*.

3.11. JCA

Distributed Work Manager Support

JBoss EAP 7.1 supports the use of distributed work managers to reschedule work execution on another work manager instance. For more information, see the Distributed Work Managers section of the *Configuration Guide*.

3.12. DATASOURCES

Flushing Datasource Connections

You can flush datasource connections using the management CLI or management console. For details, see the Flushing Datasource Connections section of the *Configuration Guide*.

Recording of Enlistment Traces is Disabled

In JBoss EAP 7.1, by default, the `enlistment-trace` attribute is set to `false` for datasources. You can enable the recording of enlistment traces by setting the `enlistment-trace` attribute to `true`.

WARNING

Enabling enlistment tracing makes tracking down errors during transaction enlistment easier, but comes with a performance impact.

3.13. RESOURCE ADAPTERS

Configuring the Generic JMS Resource Adapter

JBoss EAP 7.1 allows you to configure a generic JMS resource adapter for use with JMS providers.

Flushing Resource Adapter Connections

You can flush resource adapter connections using the management CLI. For details, see the Flushing Resource Adapter Connections section of the *Configuration Guide*.

Recording of Enlistment Traces is Disabled

In JBoss EAP 7.1, by default, the `enlistment-trace` attribute is set to `false` for resource adapters. You can enable the recording of enlistment traces by setting the `enlistment-trace` attribute to `true`.

WARNING

Enabling enlistment tracing makes tracking down errors during transaction enlistment easier, but comes with a performance impact.

3.14. EJB

Clustered Singleton MDB Support

JBoss EAP 7.1 now supports the use of clustered singleton MDBs. When an MDB is identified as a clustered singleton and deployed in a cluster, it will only be active on one node at a time. When the server node fails or is shut down, the clustered singleton MDB is activated on a different node and starts consuming messages on that node.

For more information, see Clustered Singleton MDBs in *Developing EJB Applications*.

Rebalancing of All Inbound MDB Connections

In JBoss EAP 7.0, you could use the **`rebalanceConnections`** activation configuration property for MDBs to allow for rebalancing of all inbound MDB connections when the underlying Artemis cluster topology changes.

In JBoss EAP 7.1, you can now set this behavior by using the **`rebalance-connections`** attribute in the **`pooled-connection-factory`** configurations in the **`messaging-activemq`** subsystem.

Legacy EJB Client Compatibility

JBoss EAP 7.1 ships with two EJB clients:

EJB client

The new EJB client is mostly, but not fully, backward compatible with the EJB client from JBoss EAP 7.0. This EJB client supports dynamic identity switching and remoting has been enhanced to support multiple identities over a single connection, instead of requiring a new connection per identity.

Legacy EJB client

The legacy EJB client provides full binary backward compatibility. This legacy EJB client can run with the client applications that were initially compiled using the EJB client from JBoss EAP 7.0. All the APIs that were present in the EJB client for JBoss EAP 7.0 are present in the legacy EJB client for JBoss EAP 7.1.
For more information, see Legacy EJB Client Compatibility in *Developing EJB Applications*.

EJB Client Code Simplification

In JBoss EAP 7.1, you can simplify the EJB client code when invoking the EJB server-side clustered components.

For more information, see EJB Client Code Simplification in *Developing EJB Applications*.

Configuring the EJB Client Address

In JBoss EAP 7.1, you can bind the EJB client's socket to a particular address and port. Then, the target EJB can read the source address and port of the remote client that invoked it.

For more information, see Configure the EJB Client Address in *Developing EJB Applications*.

Single artifactID for jboss-ejb-client Dependencies

Including the **`jboss-ejb-client`** dependency, with its version managed using **`wildfly-ejb-client-bom`**, includes all the required dependencies for the EJB client.

In the previous releases of JBoss EAP, the dependencies had to be included manually in the **`pom.xml`**. In JBoss EAP 7.1, this is not required.

For more information, see Project Dependencies for Remote EJB Clients in *Developing EJB Applications*.

Regular Expression Support in Interceptor Bindings

In JBoss EAP 7.1, you can set the **`allow-ejb-name-regex`** attribute of the **`ejb3`** subsystem to **`true`** to allow regular expressions in interceptor bindings. This allows interceptors to be mapped to all beans that match the specified regular expression.

For more information, see Configure a Container Interceptor in *Developing EJB Applications*.

3.15. JSF

Multi-JSF Support

JBoss EAP 7.1 provides full support for Multi-JSF. This feature enables a user to replace the JSF implementation provided with JBoss EAP with a user-supplied JSF implementation. This feature also enables a user to install multiple JSF implementations and easily switch the default implementation.

Be aware that the following issue may occur when providing and installing your own JSF implementations:

Mojarra/MyFaces 2.1.x/2.0.x

JBoss EAP 7 is a certified implementation of Java EE 7. However, if you install an alternative JSF implementation of version 2.1 or older, JBoss EAP 7 is no longer compliant with Java EE 7. These older versions are compliant with the JSF 2.0 specification defined in JSR-314, so features from the JSF 2.2 specification defined in JSR-344 will be missing.

For more information, see Multi-JSF Implementation of JavaServer Faces in the *Configuration Guide*.

3.16. HIBERNATE

Upgraded to Hibernate ORM 5.1

JBoss EAP 7.1 now includes Hibernate ORM 5.1. The Hibernate ORM 5.1 release includes many performance improvements and bug fixes. It also introduces the following new features and improvements:

Hibernate ORM 5.1 Features

- In Hibernate Query Language (HQL), you can define a join to an entity, not just a mapped association. For example:

```
select ...
from FinancialRecord f
    left join User u
        on r.lastUpdateBy = u.username
```

- In addition to providing the ability to load a single identity by identifier, the API now also supports loading multiple entities of same type by identifier by using the Hibernate native API **`Session`** interface. For example:

```
// Load Users 1, 2 and 3 at one shot
List<User> users = session.byMultipleIds(User.class).multiLoad( 1,
2, 3 );
```

- This release offers improvements in CDI integration, including solutions to the issue that occurs when Hibernate attempts to access the CDI **`BeanManager`** too soon. For more information, see HHH-8706 and HHH-10477.

- When defining an Envers audit query, you can now refer across one-to-one and many-to-one associations.

Upgraded to Hibernate Validator 5.3.x

JBoss EAP 7.1 now includes Hibernate Validator 5.3.x. Notable highlights include:

- Bug fixes
- The ability to add dynamic payloads to constraint violations
- A new programmatic API for constraint definition and declaration
- New translations of the built-in constraint messages

For more information, see New Features of Hibernate Validator 5.3.x in the *Development Guide*.

Access to Properties of Associations in Envers Queries

In JBoss EAP 7.1, you can access properties of associated entities in Envers queries. For more information, see Traversing Entity Associations Using Properties of Referenced Entities in *Developing Hibernate Applications*.

Define Lazy Loading Attribute Fetch Groups

In JBoss EAP 7.1, if you are using bytecode enhanced lazy loading, you can define the groupings of attributes to be fetched when one of the group is accessed. For more information, see Lazy Attribute Loading in *Developing Hibernate Applications*.

3.17. HIGH AVAILABILITY

New Load Balancer Profile

JBoss EAP 7.1 includes a new load balancer profile that is preconfigured to allow a server to run as a load balancer. The standalone server configuration file for this profile is `standalone-load-balancer.xml`, located in the `EAP_HOME/standalone/configuration/` directory. The managed domain profile is `load-balancer`, defined in the `EAP_HOME/domain/configuration/domain.xml` file. For information on using this profile, see Configure Undertow as a Load Balancer Using mod_cluster in the *Configuration Guide*.

3.18. RESTEASY

Display Resource Details of REST Endpoints

In JBoss EAP 7.1, you can use the `read-resource` management CLI operation on the `jaxrs` subsystem for deployments to view details about RESTEasy endpoints. For more information, see Viewing RESTEasy Endpoints in *Developing Web Services Applications*.

Jackson Module Support for Java 8

JBoss EAP 7.1 provides support for the Jackson modules needed for Java 8 features. For more information, see Jackson Module Support for Java 8 in *Developing Web Services Applications*.

JSON Filter Support

In JBoss EAP 7.1, you can annotate classes with `@JsonFilter` to perform dynamic filtering. For more information, see JsonFilter Support in RESTEasy Jackson2 in *Developing Web Services Applications*.

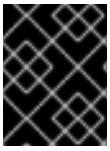
Logging RESTEasy Providers and Interceptors

RESTEasy logs the used providers and interceptors at the **DEBUG** logging level. For more information, see Logging RESTEasy Providers and Interceptors in *Developing Web Services Applications*.

3.19. MESSAGING

Messaging JDBC Persistence Store

In JBoss EAP 7.1, you can use JDBC to persist messages and binding data to a database instead of the default file-based journal.



IMPORTANT

JBoss EAP 7.1 currently supports only Oracle Database 12c and excludes high availability (HA) topologies.

For more information, see the Messaging Journal Persistence Using a JDBC Database section of *Configuring Messaging*.

Setting the Client Thread Pool Size Using System Properties

The following system properties can be used to set the size of a client's global thread pool and global scheduled thread pool.

- `activemq.artemis.client.global.thread.pool.max.size`
- `activemq.artemis.client.global.scheduled.thread.pool.core.size`

For more information, see the Client Thread Management section of *Configuring Messaging*.

Access an AMQ Broker Using the Integrated ActiveMQ Artemis Resource Adapter

You can use the integrated ActiveMQ Artemis resource adapter in the `messaging-activemq` subsystem of JBoss EAP to access an external Red Hat JBoss AMQ 7 broker.

For more information, see Using the Integrated Artemis Resource Adapter for Remote Connections in *Configuring Messaging*.

3.20. CLIENT CONFIGURATION

New Client Configuration File

JBoss EAP 7.1 introduces a `wildfly-config.xml` configuration file that allows you to specify various client configurations, such as EJB, Elytron authentication, and remoting, in a single configuration file.

See Client Configuration Using the wildfly-config.xml File in the *Development Guide* for information on the clients and types of configuration that can be done using the `wildfly-config.xml` file.

3.21. JBOSS SERVER MIGRATION TOOL

JBoss Server Migration Tool Available

The JBoss Server Migration Tool is now available with the JBoss EAP 7.1 distribution. This assists you in migrating your JBoss EAP 6.4 or 7.0 server configuration to JBoss EAP 7.1. It can convert both standalone server and managed domain configurations.

For more information on using the JBoss Server Migration Tool, see Use the JBoss Server Migration Tool to Migrate Server Configurations in the JBoss EAP *Migration Guide*.

3.22. DOCUMENTATION

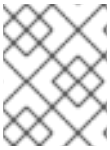
Performance Tuning Guide Available

The *Performance Tuning Guide* is now available for JBoss EAP 7.1. This guide provides optimization recommendations for common JBoss EAP use cases, as well as instructions for monitoring performance and diagnosing performance issues.

3.23. GRAPHICAL INSTALLER

Graphical Installer Provides Custom JSF Installation Option

You can install a custom JSF implementation when using the graphical installer to install JBoss EAP 7.1. On the **Configure Runtime Environment** page of the installer wizard, select **Perform advanced configuration** → **Install JSF implementation** and click **Next**. Provide the required details on the **JSF Setup** page and complete the rest of the installation.



NOTE

The JBoss EAP 7.1 installer supports installing MyFaces v2.1.x/v2.2.x and Mojarra v2.1.x/v2.2.x. The MyFaces implementation itself is not supported.

3.24. QUICKSTARTS

New Quickstart Available: ha-singleton-deployment

The `ha-singleton-deployment` quickstart is shipped with JBoss EAP 7.1. This is a complete working example of a service packaged in an application as a cluster-wide singleton using singleton deployments.

New Quickstart Available: messaging-clustering-singleton

The `messaging-clustering-singleton` quickstart is shipped with JBoss EAP 7.1. This quickstart demonstrates clustering using ActiveMQ Artemis with MDB singleton configuration.

Quickstart Updates for Elytron Security

The following quickstarts are new for JBoss EAP 7.1 and demonstrate how Elytron can be used to secure applications.

- ejb-security-context-propagation
- ejb-security-jaas
- ejb-security-programmatic-auth
- helloworld-mutual-ssl
- helloworld-mutual-ssl-secured
- helloworld-ssl

The following existing quickstarts were updated to use Elytron security:

- ejb-asynchronous
- ejb-multi-server
- ejb-remote
- ejb-security

- helloworld-jms
- servlet-security
- shopping-cart

CHAPTER 4. TECHNOLOGY PREVIEW

WARNING

The following configurations and features are provided as technology previews only. They are not supported for use in a production environment, and may be subject to significant future changes. See this note on the Red Hat Customer Portal on the support scope for Technology Preview features.

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 a technology preview feature in JBoss EAP 7.1. You can invoke EJBs over an HTTP load balancer. This can be done using the EJB/naming client APIs. For more information, see EJB Invocation Over HTTP in the *Developing EJB Applications*.

Modern Enterprise Web Applications with Server-side JavaScript on JVM

JBoss EAP 7.1 allows you to write server-side JavaScript, using JDK 8 Nashorn capabilities, to quickly develop REST endpoints that can pull in CDI beans, perform JNDI lookups, and invoke JPA entity beans. The **undertow** subsystem provides this capability as technology preview only.

Server-sent Events in Java (SSE)

An implementation of the server-sent event model in Java is provided as technology preview for users working with mobile and rich clients. This includes only the server implementation.

Configuring the Security Manager Subsystem Using the Management Console

In JBoss EAP 7.1, the ability to configure the **security-manager** subsystem from the management console is provided as technology preview only.

Download Maven Repository Using the Offliner Application

JBoss EAP 7.1 provides the ability to use the Offliner application to download the Maven repository as technology preview only. For more information, see Download the JBoss EAP Maven Repository Using the Offliner Application in the *Development Guide*.

Elytron Features

The following Elytron features are provided as technology preview only:

- Using a **filesystem-realm**, which is a simple security realm definition backed by the file system.
- Using a modifiable **custom-realm**, which is a custom security realm that implements **org.wildfly.security.auth.server.ModifiableSecurityRealm**.
- Operations for identity manipulation on a **ldap-realm** or **jdbc-realm**.

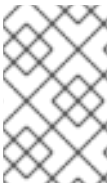
Management CLI Match Regular Expression Operator

The match regular expression (~=) operator for management CLI **if-else** control flow is provided as technology preview only. For more information, see Use if-else Control Flow in the *Management CLI Guide*.

CHAPTER 5. UNSUPPORTED AND DEPRECATED FUNCTIONALITY

5.1. 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 7.1.



NOTE

The unsupported features listed in the Unsupported Features section of the *7.0.0 Release Notes* also apply to the JBoss EAP 7.1 release, unless they are mentioned in the New Features and Enhancements section of this document.

Messaging (ActiveMQ Artemis)

The following messaging features are not supported in JBoss EAP 7.1:

- AMQP, STOMP, REST, MQTT, and OpenWire protocols
- Netty over HTTP and Netty Servlet transport options for connectors/acceptors
- OIO (Old Java IO) connectors/acceptors type is no longer possible to configure
- Vert.x, AeroGear, Spring and Jolokia integration
- Dynamic queue creation
- Chain cluster
- Clustered message grouping
- Using ActiveMQ Artemis Management using JMX
- Graceful shutdown/scaling down of nodes in an Artemis cluster
- Colocated HA topology configured using replication-colocated/shared-store-colocated

NOTE

However, colocated HA topology is supported as described in the Colocated Backup Servers section of *Configuring Messaging*.

- Using messaging with MAPPED journal type
- Avoiding network isolation
- Configuring multiple cluster connections

Infinispan APIs

Infinispan is delivered as a private module to provide the caching capabilities of JBoss EAP. Infinispan is not supported for direct use by applications.

Jackson API

The Jackson 1 libraries are not supported for development or production use in JBoss EAP.

NOTE

The Jackson 2 libraries are supported. For more information, see Does JBoss EAP support the use of Jackson libraries? on the Red Hat Customer Portal.

OAuth with RESTEasy
OAuth is not supported with RESTEasy.

ElytronAuthenticator
It is unsupported to use the `ElytronAuthenticator` class to propagate security identities. For more information, see Using the ElytronAuthenticator to Propagate Identities in *How to Configure Identity Management*.

5.2. DEPRECATED FEATURES

Some features have been deprecated with the release of JBoss EAP 7.1. This means that no enhancements will be made to these features, and they may be removed in the future.

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.

Platforms and Frameworks

Java Virtual Machine

- HP-UX

Operating Systems and Related Web Servers

- Windows Server 2008 and associated IIS web server
- Solaris 10 / 11 and associated web servers
- HP-UX
- RHEL 6 32 bit

Databases

- IBM DB2 e9.7
- MySQL 5.5
- Microsoft SQL Server 2012
- PostgreSQL 9.3
- Enterprise DB Postgres Plus Advanced Server 9.3
- Sybase 15

JMS Providers/Adapters

- IBM WebSphere MQ 7.5
- TIBCO EMS

LDAP Servers

- RHEL Directory Server 9.1
- Microsoft Active Directory 2008

Tested Frameworks

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

Cloud

- Amazon EC2
- Microsoft Azure

JBoss EAP Container Image

The JBoss EAP base image for containers, **`registry.access.redhat.com/jboss-eap-7-tech-preview/eap70`**, distributed through the Red Hat Docker Registry will not be updated for JBoss EAP 7.1 and this image will be removed for the JBoss EAP 7.1 release.

Attributes

NOTE

In most cases, deprecated attributes are not shown in the management console.

- The following attributes for HTTP listeners in the **`undertow`** subsystem are deprecated:
  - enable-spdy
  - enabled
  - enabled-cipher-suites
  - enabled-protocols
  - security-realm
  - ssl-session-cache-size
  - ssl-session-timeout
  - verify-client
- The following attributes for caches in the **`infinispan`** subsystem are deprecated:
  - queue-flush-interval

    - queue-size

- The following attributes in the `iiop-openjdk` subsystem are deprecated:
    - add-component-via-interceptor
    - queue-flush-interval
- The following attributes of the `remote-outbound-connection` resource in the `remoting` subsystem are deprecated:
    - protocol
    - security-realm
    - username

Resources

- The following core management resources are deprecated because management security is now provided by Elytron.
    - audit
    - ldap-connection
    - security-realm
- The following remoting outbound connections in the `remoting` subsystem are deprecated:
    - local-outbound-connection
    - outbound-connection
- The following persistent store types in the `infinispan` subsystem are deprecated:
    - binary-jdbc
    - mixed-jdbc

Operations

- The following management operation for the `jaxrs` subsystem is deprecated:
    - show-resources

CHAPTER 6. RESOLVED ISSUES

See Resolved Issues for JBoss EAP 7.1.0 to view the list of issues originating from customer cases that have been resolved for this release.

CHAPTER 7. FIXED CVES

JBoss EAP 7.1 includes fixes for the following security related issues:

- CVE-2016-6311: Internal IP address disclosed on redirect when request header Host field is not set
- CVE-2016-2141: Add authorization checks by default on JGroups message receipt
- CVE-2016-5406: RBAC configurations are discarded by transformers for legacy slaves running management API versions 1.8 and earlier
- CVE-2016-4993: HTTP header injection / response splitting
- CVE-2015-0254: XXE and RCE via XSL extension in JSTL XML tags
- CVE-2016-7046: Long URL proxy request lead to java.nio.BufferOverflowException and DoS
- CVE-2016-8627: Potential EAP resource starvation DOS attack via GET requests for server log files
- CVE-2016-7061: Sensitive data can be exposed at the server level in domain mode
- CVE-2016-8656: unsafe chown of server.log in jboss init script allows privilege escalation
- CVE-2016-9589: ParseState headerValuesCache can be exploited to fill heap with garbage
- CVE-2017-2595: Arbitrary file read via path traversal
- CVE-2016-9606: Resteasy: Yaml unmarshalling vulnerable to RCE
- CVE-2017-2666: HTTP Request smuggling vulnerability due to permitting invalid characters in HTTP requests
- CVE-2017-2670: Websocket non clean close can cause IO thread to get stuck in a loop
- CVE-2016-4978: JMSObjectMessage deserializes potentially malicious objects allowing Remote Code Execution
- CVE-2017-7525: jackson-databind: Deserialization vulnerability via readValue method of ObjectMapper
- CVE-2017-2582: SAML request parser replaces special strings with system properties
- CVE-2014-9970: jasypt: Vulnerable to timing attack against the password hash comparison
- CVE-2015-6644: bouncycastle: Information disclosure in GCMBlockCipher
- CVE-2017-5645: log4j: Socket receiver deserialization vulnerability
- CVE-2017-7536: hibernate-validator: Privilege escalation when running under the security manager
- CVE-2017-12165: Improper whitespace parsing leading to potential HTTP request smuggling
- CVE-2017-7559: Potential http request smuggling as Undertow parses the http headers with unusual whitespaces

- CVE-2016-7066: World executable permission on bin/jboss-cli after installation. Any users of the system could cause harm, or shutdown the running instance of JBoss EAP
- CVE-2017-12167: Wrong privileges on multiple property files

CHAPTER 8. KNOWN ISSUES

See Known Issues for JBoss EAP 7.1.0 to view the list of known issues for this release.

Additionally, be aware of the following:

- The **`jboss-jaxrpc-api_1.1_spec`** package lists an incorrect license in the JBoss EAP **`licenses.xml`** file. The correct license information is CDDL or GPLv2 with the Classpath Exception.
- There are some discrepancies in the artifact licenses between the RPM and the ZIP installation. The license information from the ZIP installation is valid, except for the **`jboss-jaxrpc-api_1.1_spec`** package license information mentioned in the previous bullet.
- There is an issue when attempting to use a credential store of type PKCS12 with the IBM JDK or HP JDK. The workaround is to use a JCEKS credential store. For more information, see JBEAP-13586.
- The following JIRAs are caused by JDK bugs and fixing them is out of scope for JBoss EAP:
  - JBEAP-8207: Elytron, IBM java, SPNEGO continuation required situation
  - JBEAP-10483: HTTP2 via JSSE and wildfly ALPN hack ssl engine is broken on Solaris 11 JBoss EAP works around this issue by disabling the OracleUcrypto provider in the default JBoss EAP configuration. However, this might cause issues on a Solaris 10 platform with HTTP over TLS. If you encounter issues, either enable the OracleUcrypto provider or update your Solaris 10 machine with the 150401-52 patch or newer.

Revised on 2018-06-27 09:31:20 EDT