



Open Liberty 2020

Release Notes for Open Liberty 20.0.0.10 on Red Hat OpenShift Container Platform

Release Notes for Open Liberty 2020 on Red Hat OpenShift Container Platform

Open Liberty 2020 Release Notes for Open Liberty 20.0.0.10 on Red Hat OpenShift Container Platform

Release Notes for Open Liberty 2020 on Red Hat OpenShift Container Platform

Legal Notice

Copyright © 2020 IBM Corp

Code and build scripts are licensed under the Eclipse Public License v1 Documentation files are licensed under Creative Commons Attribution-NoDerivatives 4.0 International (CC BY-ND 4.0)

Abstract

These release notes contain the latest information about new features, enhancements, fixes, and issues contained in Open Liberty 2020 on Red Hat OpenShift Container Platform release.

Table of Contents

CHAPTER 1. FEATURES	3
1.1. RUN YOUR APPS USING 20.0.0.10	3
1.1.1. Install a signed certificate with Automatic Certificate Management Environment Support 2.0 (acmeCA-2.0)	3
1.2. SIGNIFICANT BUGS FIXED IN THIS RELEASE	4
1.2.1. Enabling openTracing with no tracer class configured	4
1.2.2. Readiness check reports UP when application fails to start	4
1.2.3. Support IIOP transmission of Supplemental Multilingual Plane characters (such as emoji) in (wide) Strings	5
1.2.4. PostgreSQL tables are not automatically generated for transaction recovery	5
1.2.5. Kafka connector can report failure for acknowledgements which eventually succeed	5
CHAPTER 2. RESOLVED ISSUES	6
CHAPTER 3. FIXED CVES	7
CHAPTER 4. KNOWN ISSUES	8

CHAPTER 1. FEATURES

With Open Liberty 20.0.0.10 you can now install a certificate signed by a Certificate Authority (CA) using the Automatic Certificate Management Environment (ACME) protocol for improved testing or user experience. Also in this release there are a number of significant bug fixes.

In [Open Liberty 20.0.0.10](#):

- [Install a signed certificate with Automatic Certificate Management Environment Support 2.0 \(acmeCA-2.0\)](#)
- [Significant bugs fixed in this release](#)

View the list of fixed bugs in [20.0.0.10](#).

1.1. RUN YOUR APPS USING 20.0.0.10

If you're using [Maven](#), here are the coordinates:

```
<dependency>
  <groupId>io.openliberty</groupId>
  <artifactId>openliberty-runtime</artifactId>
  <version>20.0.0.10</version>
  <type>zip</type>
</dependency>
```

Or for [Gradle](#):

```
dependencies {
  libertyRuntime group: 'io.openliberty', name: 'openliberty-runtime', version: '[20.0.0.10,)'
}
```

Or if you're using Docker:

```
FROM open-liberty
```

1.1.1. Install a signed certificate with Automatic Certificate Management Environment Support 2.0 (acmeCA-2.0)

For default Transport Security (SSL/TLS) support, Open Liberty provides a self-signed certificate. With the Automatic Certificate Management Environment (ACME) Support 2.0 feature, a Certificate Authority (CA) signed certificate can be installed, providing a trusted certificate for an improved testing or user experience. The self-signed certificate from Open Liberty allows you to enable transport security right away, but most browsers will mark the certificate as insecure and provide a warning or error to the user when accessing a website with a self-signed certificate. Although a CA signed certificate solves this problem, it can be costly and may not be available during development and testing. Getting a CA signed certificate (for example, Lets Encrypt), provides a middle ground of a trusted certificate.

With the Automatic Certificate Management Environment (ACME) Support 2.0 feature, provide the directory URI for the ACME CA and the domain name for the server. Public ACME providers call back on port 80 to verify domain ownership before issuing a certificate. When the Open Liberty server starts, the provided CA directory URI will be used to request a certificate. The CA signed certificate is installed in the keystore and acts as the default certificate.

Add the feature to the **server.xml**:

```
<featureManager>
  <feature>acmeCA-2.0</feature>
</featureManager>

<acmeCA directoryURI="https://acme.host.com/directory" >
  <domain>theDomainThatIOwn.com</domain>
  <accountContact>mailto:my_email_addr@theDomainThatIOwn.com</accountContact>
</acmeCA>

<httpEndpoint host="*" httpPort="80" httpsPort="443" id="defaultHttpEndpoint"/>
<keyStore password="password_for_keystore" id="defaultKeyStore"/>
```

For more information:

- [High level summary of the ACME protocol](#)
- [The ACME spec](#)
- [Popular public ACME CA](#)

1.2. SIGNIFICANT BUGS FIXED IN THIS RELEASE

We've spent some time fixing bugs. The following sections describe just some of the issues we resolved in this release. If you're interested, here's the full list of [fixed bugs in 20.0.0.10](#).

1.2.1. Enabling openTracing with no tracer class configured

When the MicroProfile 3.x feature is enabled, **mpOpenTracing-1.3** is also enabled even if you have no intention of using OpenTracing. Most likely the tracer fails to load because it is not added to the classpath. Previously, **mpOpenTracing-1.3** code tried to load the tracer on every JAX-RS request. This logic has been improved by loading the tracer only once to improve efficiency. Subsequent requests do not load the tracer again. Also the tracer loaded for the application will be removed when the application stops.

Resolved in issue [12613](#).

1.2.2. Readiness check reports UP when application fails to start

The MicroProfile Health Check reports **UP** when application fails to start. This bug was fixed by changing the implementation to test for the following:

- Proper state changes when a failing to start app is deployed before server start.
- Proper state changes when apps are deployed before server start.
- Proper state changes when apps are dynamically deployed after server start (including delayed apps).
- Proper state changes from success to failure, when a subsequent app fails health checks.
- No effect on a failed health check state when a subsequent app succeeds health checks.

Resolved in issue [11722](#).

1.2.3. Support IOP transmission of Supplemental Multilingual Plane characters (such as emoji) in (wide) Strings

Have you ever wished you could send 🍌 in your IOP transmission? Probably not, but now you can! All thanks to the update of Yoko ORB which was used by Open Liberty to one that supports the transmission of Supplemental Multilingual Plane Unicode text characters, which are encoded as two 16-byte values.

Resolved in issue [13613](#).

1.2.4. PostgreSQL tables are not automatically generated for transaction recovery

It's possible to configure Liberty to use a database for its transaction logs as [documented](#). PostgreSQL used to be one of the databases for which the necessary tables had to be created manually using the documented DDL as a guide. Now, Liberty can automatically create these tables for PostgreSQL.

Resolved in issue [13817](#).

1.2.5. Kafka connector can report failure for acknowledgements which eventually succeed

When **Message.ack()** is called, the Kafka connector returns a result as a **CompletionStage** which completes when the Kafka commit operation has completed successfully. In some cases, the Kafka commit operation can fail and report a retrieable exception, for example if there is a temporary problem contacting the Kafka broker. Previously, the Kafka connector would report the retrieable exception via the **CompletionStage**, but if it then went to run a commit operation for a later offset, the earlier messages would actually be successfully committed. With this fix, if a Kafka commit operation fails with a retrieable exception, the Kafka connector will retry the commit operation as necessary and will not report the exception via the **CompletionStage**. If the commit operation eventually succeeds, the success will be reported via the **CompletionStage**.

Find out more about message acknowledgement in reactive systems in [our new reactive messaging guide](#).

Resolved in issue [13404](#).

CHAPTER 2. RESOLVED ISSUES

See the [Open Liberty 20.0.0.10 issues that were resolved for this release](#) .

CHAPTER 3. FIXED CVES

For a list of CVEs that were fixed in Open Liberty 20.0.0.10, see [security vulnerabilities](#).

CHAPTER 4. KNOWN ISSUES

See the [list of issues that were found but not fixed during the development of 20.0.0.10](#) .