



Red Hat JBoss Developer Studio 10.1 Release Notes

Highlighted features in 10.1

Misha Husnain Ali Supriya Bharadwaj
Red Hat Developer Group Documentation
Team

Red Hat JBoss Developer Studio 10.1 Release Notes

Highlighted features in 10.1

Misha Husnain Ali
mhusnain@redhat.com

Supriya Bharadwaj
sbharadw@redhat.com

Legal Notice

Copyright © 2017 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document lists and briefly describes new and improved features of Red Hat JBoss Developer Studio 10.1.

Table of Contents

CHAPTER 1. INTRODUCTION TO RED HAT JBOSS DEVELOPER STUDIO	3
1.1. ABOUT RED HAT JBOSS DEVELOPER STUDIO	3
1.2. USE CASES OF JBOSS DEVELOPER STUDIO	3
1.3. ABOUT THIS RELEASE	5
1.4. ECLIPSE AUTOMATED REPORTING INTERFACE (AERI)	5
1.5. NEW FEATURES AND ENHANCEMENTS	5
1.6. ISSUES	19
1.7. KNOWN ISSUES	19
1.8. APPLY THIS RELEASE	20

CHAPTER 1. INTRODUCTION TO RED HAT JBOSS DEVELOPER STUDIO

1.1. ABOUT RED HAT JBOSS DEVELOPER STUDIO

Red Hat JBoss Developer Studio is a set of Eclipse-based development tools. It contains plug-ins that integrate with Eclipse to extend the existing functionality of the integrated development environment (IDE).

JBoss Developer Studio is designed to increase your productivity when developing applications. You can focus on building, testing and deploying your applications because JBoss application development tools are integrated in one IDE. Furthermore, JBoss Developer Studio can assist your application development with its unique features in the following ways:

- ✧ Develop new applications using the wizards and project examples of Red Hat Central
- ✧ Add powerful functionality to applications with minimal effort using Forge Tools
- ✧ Build web interfaces with ease using the visual editing and drag-and-drop utilities of Visual Web Tools and Mobile Web Tools
- ✧ Experience browsers automatically refreshing in response to modified application resources with LiveReload Tools
- ✧ Incorporate JSF, Seam, JAX-RS, Hibernate, CDI and other popular APIs into applications with simplicity using the tool-driven interface
- ✧ Preview and test mobile web applications on a variety of simulated mobile devices using BrowserSim
- ✧ Create, build and test Cordova-based hybrid mobile applications for iOS and Android platforms using Hybrid Mobile Tools and CordovaSim
- ✧ Deploy applications to JBoss runtime servers and the cloud using JBoss Server Tools and OpenShift Tools

JBoss Developer Studio comes built around Eclipse and packaged with all of the necessary dependencies and third-party plug-ins for simplified installing. For developers already running Eclipse, JBoss Developer Studio can also be installed through Eclipse Marketplace. For distinction, this latter JBoss Developer Studio installation is called JBoss Developer Studio BYOE (Bring Your Own Eclipse).

1.2. USE CASES OF JBOSS DEVELOPER STUDIO

JBoss Developer Studio assists Java EE developers by integrating JBoss technology and APIs in a single development environment. Here are a few ways that JBoss Developer Studio makes the work of a developer easier:

1.2.1. Web applications

Red Hat Central provides wizards that generate skeleton and sample projects, enabling you to focus on developing the functionality of your applications. The wizards create web applications based on different APIs and technologies, showing the usage and advantages of each.

JBoss Developer Studio also offers project file templates in a range of popular programming languages, including HTML, XHTML, and JSF.

Palettes in JBoss Developer Studio give access to the core elements of the JSF, RichFaces and Seam APIs, for use in developing the user interfaces of your applications. Elements of these APIs can be dragged and dropped directly into your project so that you can create richer user interfaces quickly. Visual Web Tools offers graphical and source viewing of files and defaults to dedicated editors for different file types. JBoss Developer Studio supports the Java EE specification and provides tools for JAX-RS, Hibernate, and CDI APIs so you can develop the server-side components of your applications effortlessly.

LiveReload Tools automatically refreshes browsers of local or deployed applications as you modify project resources to save you from needing to manually refresh. You can even experience automatic refreshing when viewing applications in browsers on external and mobile devices, with application web addresses easy to navigate to with QR codes.

1.2.2. Web applications optimized for mobile devices

Mobile Web Tools provides support for HTML5 and jQuery Mobile to enable you to create web applications optimized across desktop and mobile clients. The HTML5 Project wizard in Red Hat Central generates a sample application using HTML5 and jQuery Mobile technologies and, together with HTML5 and jQuery Mobile project file templates, helps you to get up and running with these APIs and technologies quickly. HTML5 and jQuery Mobile widgets can be dragged from the jQuery Mobile palette into your project files and, in conjunction with the widget wizards, enable you to effortlessly develop customized user interfaces for your mobile web applications.

BrowserSim allows you to view your web applications on a variety of simulated mobile devices so that you can ensure they will be correctly formatted. LiveReload also extends to BrowserSim allowing you to experience automatic browser refreshing as you develop your mobile web applications. The integration of Firebug Lite and Weinre capabilities with BrowserSim assists you to inspect the page source of web pages with familiar tools.

1.2.3. Hybrid mobile applications

Hybrid Mobile Tools provides support for developing and building Cordova-based hybrid mobile applications for iOS and Android platforms. The Hybrid Mobile application wizard assists you to quickly generate new projects, while the Cordova Configuration Editor and Cordova Plug-in Discovery wizard help you to efficiently modify the capabilities of your projects. Hybrid Mobile Tools provides actions that simplify your workflow, for example calling your system installed Android and iOS SDKs from within the IDE to emulate or run your hybrid mobile applications. With wizards to export workspace projects to Cordova-enabled native projects or ready-to-sign applications, you can quickly be ready to share your hybrid mobile projects and applications.

CordovaSim enables you to view and test your hybrid mobile applications on Android and iOS simulated mobile devices so that you can ensure they look and work as expected. You can interact with your mobile applications through BrowserSim and use the device input panel to provide sample data to your applications for device functions like cameras and accelerometers. An advantage of CordovaSim is that it does not require native SDKs to be installed on your system, unlike native SDK emulators. Additionally, by teaming the device control panel with BrowserSim, you get all of the great functionality of BrowserSim, such as skins and LiveReload, while simulating your hybrid mobile applications.

1.2.4. Applications for cloud deployment

OpenShift Tools deploys your applications directly to the cloud on the Red Hat OpenShift platform. You can create and manage your OpenShift account and manage the deployment of applications to OpenShift within the IDE. In addition to using the OpenShift Application wizard to create and deploy new OpenShift applications, OpenShift Tools can import applications already deployed on

OpenShift so that you can further develop them and manage their deployment from the comfort of the IDE.

1.3. ABOUT THIS RELEASE

JBoss Developer Studio 10.1 is an upgrade of JBoss Developer Studio 10.0. The upgrade has the following purposes:

- ✦ It includes Eclipse Neon
- ✦ It includes Red Hat JBoss EAP 7 in the bundled installer.
- ✦ It requires a minimum of Java 8 to run.
- ✦ It introduces new features, which are outlined in the New Features section.
- ✦ It contains new features for the existing tools.
- ✦ It resolves issues identified in earlier versions of JBoss Developer Studio.

For more information about operating systems, chip architectures and Java developer kits supported by this release, see <https://access.redhat.com/site/articles/427493> on the Red Hat Customer Portal.

1.4. ECLIPSE AUTOMATED REPORTING INTERFACE (AERI)

As of August 1, 2016, the JBoss Tools Community Acceptance Testing (JBoss Tools CAT) has suspended operations. Many thanks to everyone who participated!

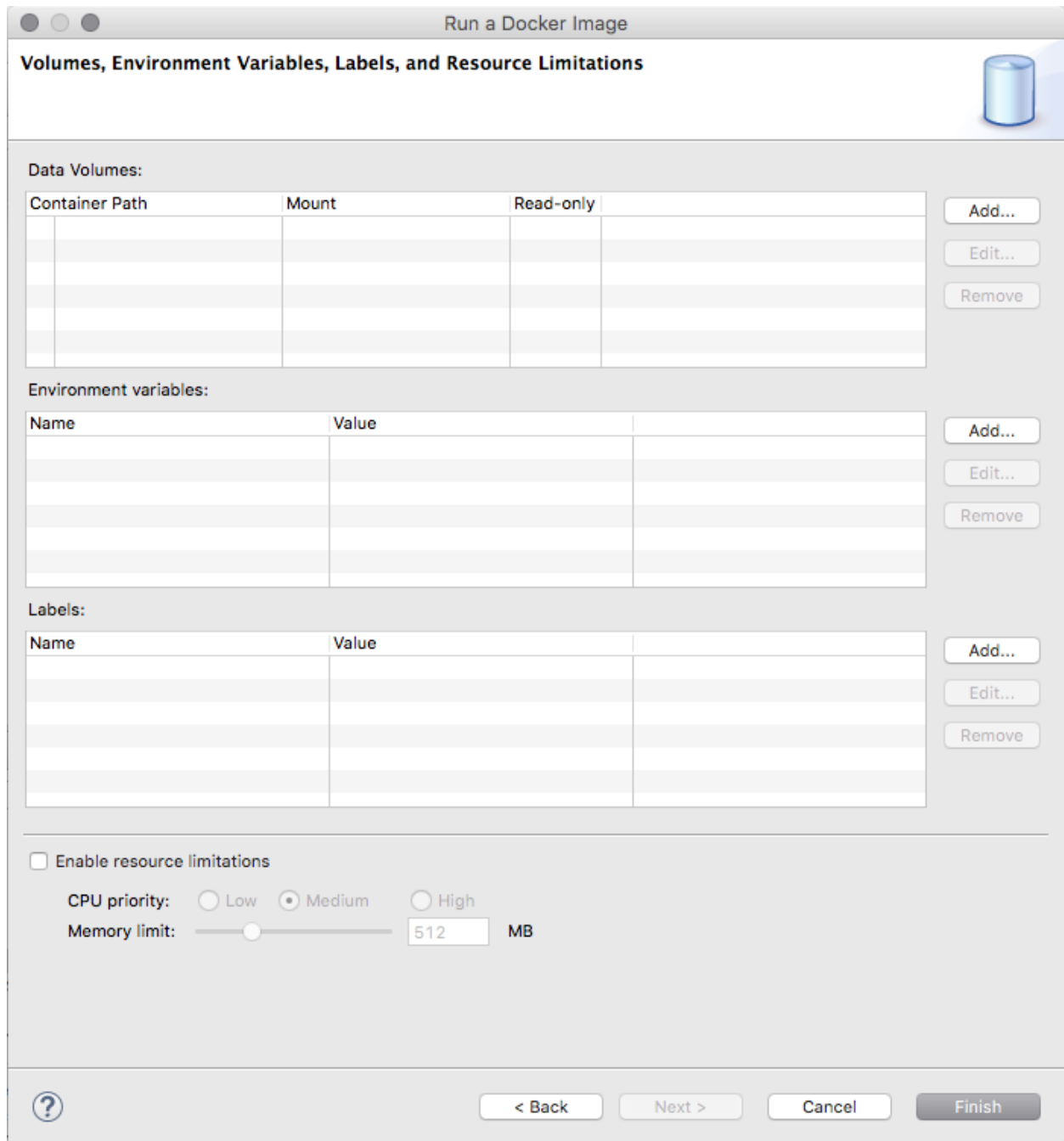
In place of the JBoss Tools CAT, we'd like to ask everyone who would like to contribute issues to JBoss Tools to enable the Eclipse Automated Reporting Interface (AERI) in JBoss Tools. You can read about configuring error reporting in JBoss Tools here: <http://tools.jboss.org/usage/#error-reporting>

1.5. NEW FEATURES AND ENHANCEMENTS

1.5.1. Docker Tools

1.5.1.1. Support for Container Labels

Users can now specify labels when running a container. The labels are saved in the launch configuration and can also be edited before relaunching the container.



See issues [498166](#), [497777](#), [485913](#) and [485936](#) in Eclipse Bugzilla for more details.

1.5.1.2. Automatically detect known Docker daemon connections

When the **Docker Explorer** view is opened, the list of existing connections (saved from a previous session) is reloaded. Additionally, the view finds new connections using default settings such the `unix:///var/run/docker.sock` Unix socket or the `DOCKER_HOST`, `DOCKER_CERT_PATH` and `DOCKER_TLS_VERIFY` environment variables. This means that by default, in a new workspace, if a Docker daemon is reachable using one of those methods, the user does not have to use the **New Connection** wizard to get a connection.

See issue [497101](#) in Eclipse Bugzilla for more details.

1.5.1.3. Extension point for Docker daemon connection settings

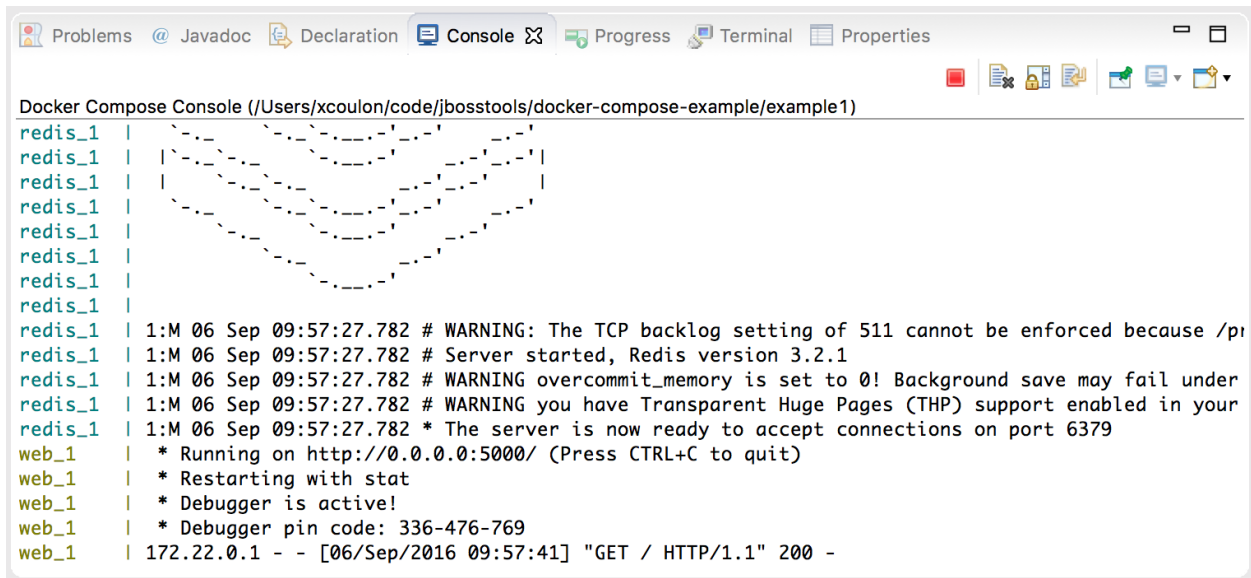
An extension point has been added to the Docker core plugin to allow for custom connection settings provisioning.

See issue [497100](#) in Eclipse Bugzilla for more details.

1.5.1.4. Support for Docker Compose

Support for Docker Compose is now available. Users can select a **docker-compose.yml** file and start Docker Compose from the context menu, using the **Run > Docker Compose** launcher shortcut.

The Docker Compose process displays its logs (with support for text coloring based on ANSI escape codes) and provides a **stop** button to stop the underlying process.



```

Docker Compose Console (/Users/xcoulon/code/jbosstools/docker-compose-example/example1)
redis_1 | _____
redis_1 | |
redis_1 | |
redis_1 | |
redis_1 | |
redis_1 | |
redis_1 | |
redis_1 | |
redis_1 | 1:M 06 Sep 09:57:27.782 # WARNING: The TCP backlog setting of 511 cannot be enforced because /p
redis_1 | 1:M 06 Sep 09:57:27.782 # Server started, Redis version 3.2.1
redis_1 | 1:M 06 Sep 09:57:27.782 # WARNING overcommit_memory is set to 0! Background save may fail under
redis_1 | 1:M 06 Sep 09:57:27.782 # WARNING you have Transparent Huge Pages (THP) support enabled in your
redis_1 | 1:M 06 Sep 09:57:27.782 * The server is now ready to accept connections on port 6379
web_1 | * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
web_1 | * Restarting with stat
web_1 | * Debugger is active!
web_1 | * Debugger pin code: 336-476-769
web_1 | 172.22.0.1 - - [06/Sep/2016 09:57:41] "GET / HTTP/1.1" 200 -

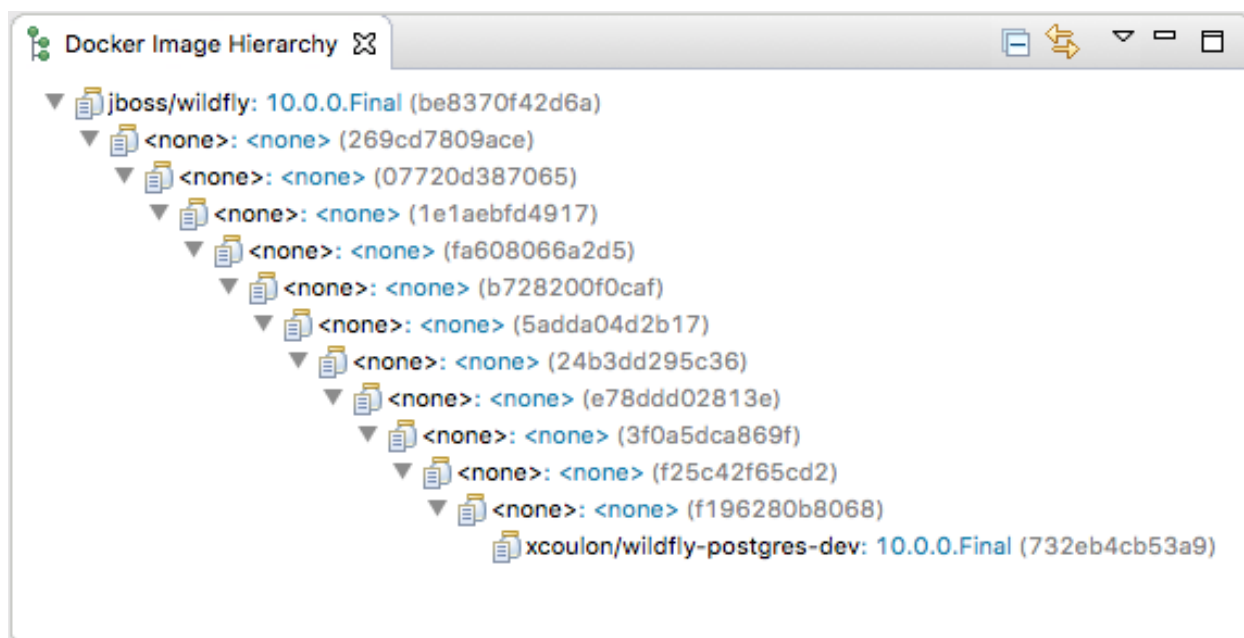
```

Also, as with the support for building and running containers, a launch configuration is created after the first call to Docker Compose on the selected **docker-compose.yml** file.

See issue [479592](#) in Eclipse Bugzilla for more details.

1.5.1.5. Docker Image Hierarchy View

The new **Docker Image Hierarchy** view lets the user view the layers for a selected image in the **Docker Explorer** view. This is useful when an image was built locally, as it helps understanding on which layers the top-level image depends.

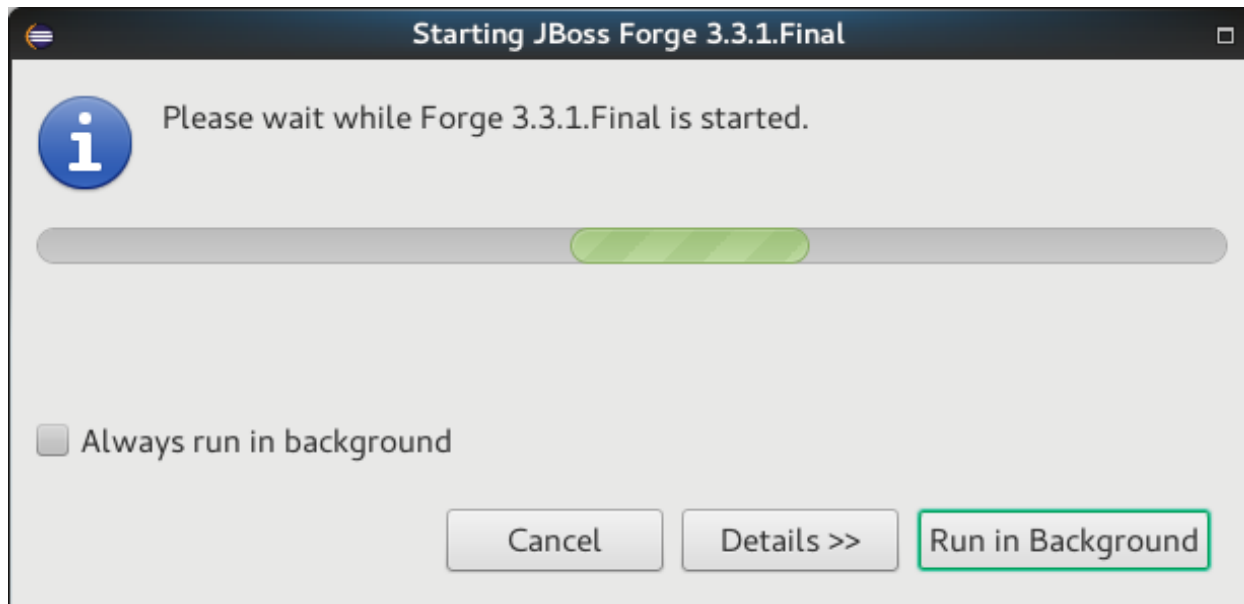


See issue [496308](#) in Eclipse Bugzilla for more details.

1.5.2. Forge Tools

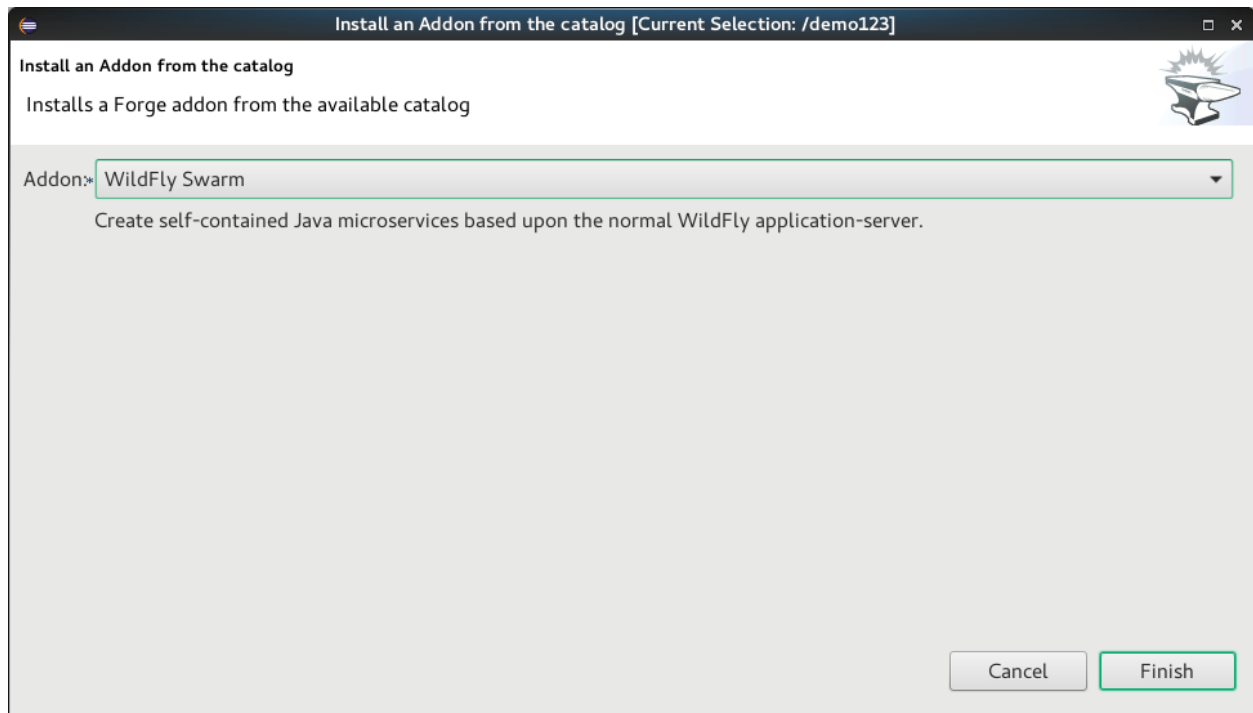
1.5.2.1. Forge Runtime updated to 3.3.1.Final

The included Forge runtime is now 3.3.1.Final. Read the official announcement [here](#).



1.5.2.2. Added Install addon from the catalog command

From Forge 3.3.0.Final onwards it is now possible to query and install add ons listed in the [Forge add ons page](#).



1.5.3. Freemarker

1.5.3.1. Freemarker 2.3.25

The Freemarker library included in the Freemarker IDE was updated to the latest available version 2.3.25.

Related JIRA: [JBIDE-22700](#)

1.5.3.2. flth/fltx file extensions added

The new **flth** and **fltx** extensions are added and associated with Freemarker IDE. **flth** refers to HTML content and **fltx** refers to XML content.

Related JIRA: [JBIDE-22636](#)

1.5.3.3. Overhaul of the plugin template parser

The parser that FreeMarker IDE uses to extract IDE-centric information (needed for syntax highlighting, related tag highlighting, auto-completion, outline view, etc.) was overhauled.

Several bugs were fixed, and support for the newer template language features were added. Also, the syntax highlighting is now more detailed inside expressions.

Related JIRAs: [JBIDE-20180](#), [JBIDE-22574](#), [JBIDE-21819](#)

1.5.3.4. Related tag background highlighting fixed

Fixed the issue when the (by default) yellow highlighting of the related FTL tags shift away from under the tag as you type.

Related JIRA: [JBIDE-22656](#)

1.5.3.5. Showing whitespace, block selection mode

The standard "Show whitespace characters" and "Toggle block selection mode" icons are now available when editing a template.

Related JIRA: [JBIDE-22751](#)

1.5.3.6. Improved automatic finishing of FreeMarker constructs

When you type symbols such as `<#`, `<@`, `${`, and `#{`, the Freemarker editor now automatically closes them.

Related JIRA: [JBIDE-22749](#)

1.5.3.7. Error positions links on the console

When a FreeMarker exception is printed to the console, the error position in it is a link that navigates to the error. This has worked long ago, but was broken for quite a while.

Related JIRA: [JBIDE-22701](#)

1.5.3.8. Fixed auto-indentation

When hitting enter, sometimes the new line haven't inherited the indentation of the last line.

Related JIRA: [JBIDE-20512](#)

1.5.3.9. Updated the "database" used for auto completion

Auto completion now knows all directives and "built-ins" up to FreeMarker 2.3.25.

Related JIRAs: [JBIDE-20386](#), [JBIDE-22765](#)

1.5.4. Hibernate Tools

1.5.4.1. Hibernate Runtime Provider Updates

A number of additions and updates have been performed on the available Hibernate runtime providers.

1.5.4.1.1. New Hibernate 5.2 Runtime Provider

With final releases available in the Hibernate 5.2 stream, the time was right to make available a corresponding Hibernate 5.2 runtime provider. This runtime provider incorporates Hibernate Core version 5.2.2.Final and Hibernate Tools version 5.2.0.Beta1.

Hibernate Configuration File (cfg.xml)

This wizard creates a new configuration file to use with Hibernate.

Container:	/hib52/src/main/java
File name:	hibernate.cfg.xml
Hibernate version:	<div> <div>✓ 5.2</div> <div>5.1</div> <div>5.0</div> </div>
Session factory name:	

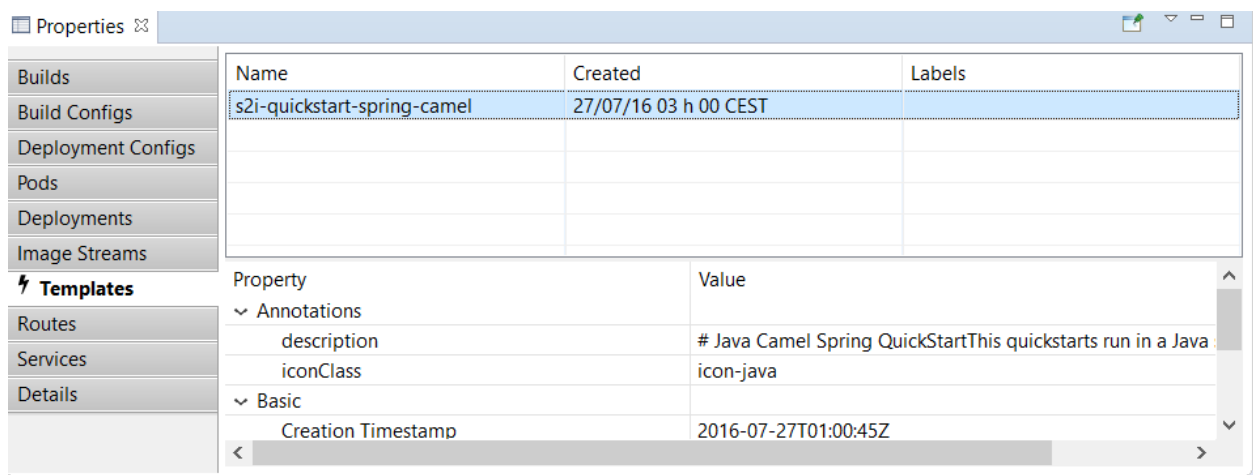
1.5.4.1.2. Other Runtime Provider Updates

- ✎ The Hibernate 4.3 runtime provider now incorporates Hibernate Core version 4.3.11.Final and Hibernate Tools version 4.3.5.Final.
- ✎ The Hibernate 5.0 runtime provider now incorporates Hibernate Core version 5.0.10.Final and Hibernate Tools version 5.0.2.Final.
- ✎ The Hibernate 5.1 runtime provider now incorporates Hibernate Core version 5.1.1.Final and Hibernate Tools version 5.1.0.CR1.

1.5.5. OpenShift

1.5.5.1. Server templates can now be displayed/edited

Server templates are now displayed in the property view under the **Templates** tab:

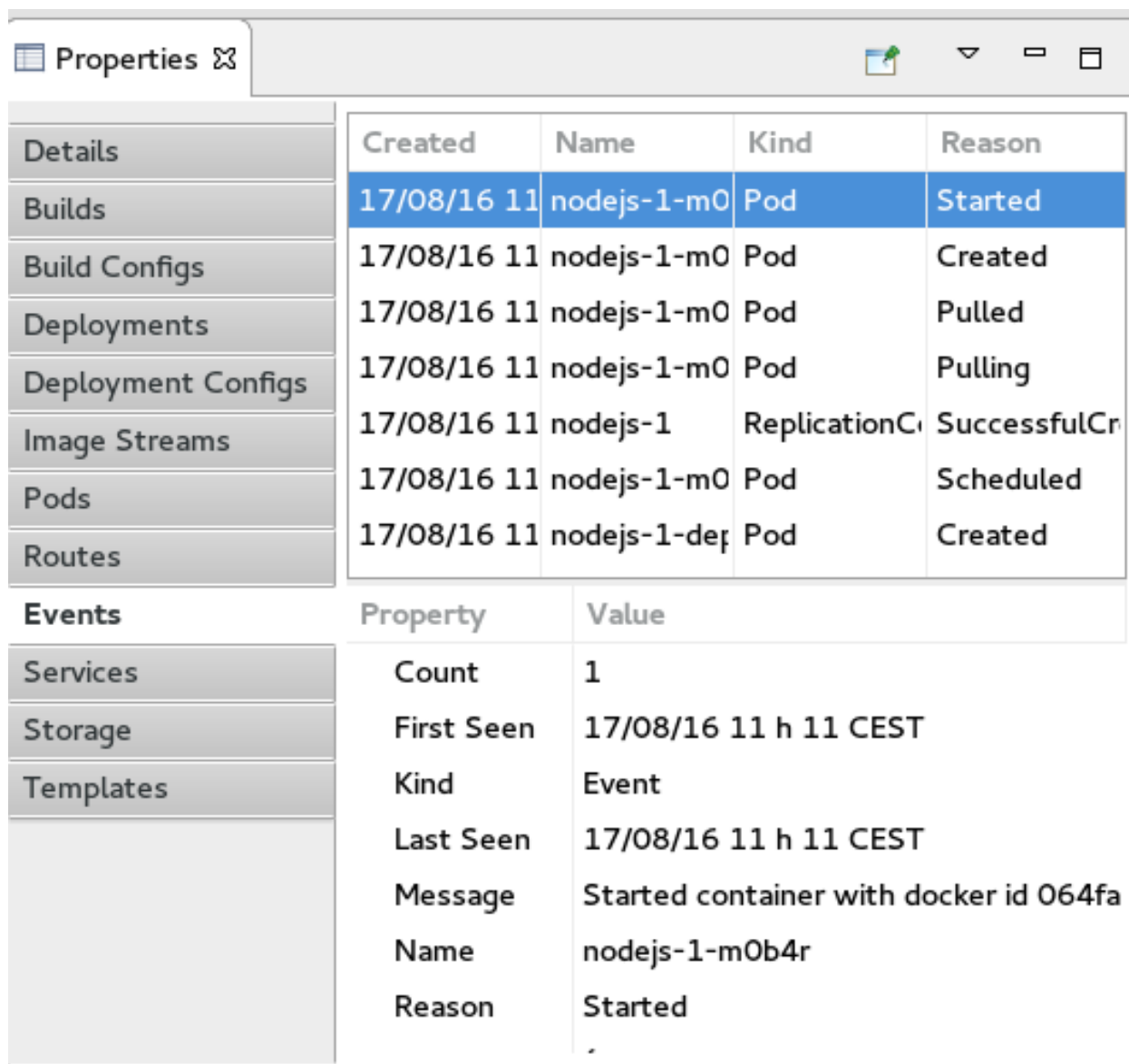


You can access/edit the content of the template with the **Edit** option.

Related JIRA: [JBIDE-22457](#)

1.5.5.2. Events can now be displayed

Events generated as part of the application life cycle now display in the **Properties** view under the **Events** tab (available at the project level):



Created	Name	Kind	Reason
17/08/16 11	nodejs-1-m0	Pod	Started
17/08/16 11	nodejs-1-m0	Pod	Created
17/08/16 11	nodejs-1-m0	Pod	Pulled
17/08/16 11	nodejs-1-m0	Pod	Pulling
17/08/16 11	nodejs-1	ReplicationC	SuccessfulCr
17/08/16 11	nodejs-1-m0	Pod	Scheduled
17/08/16 11	nodejs-1-dep	Pod	Created

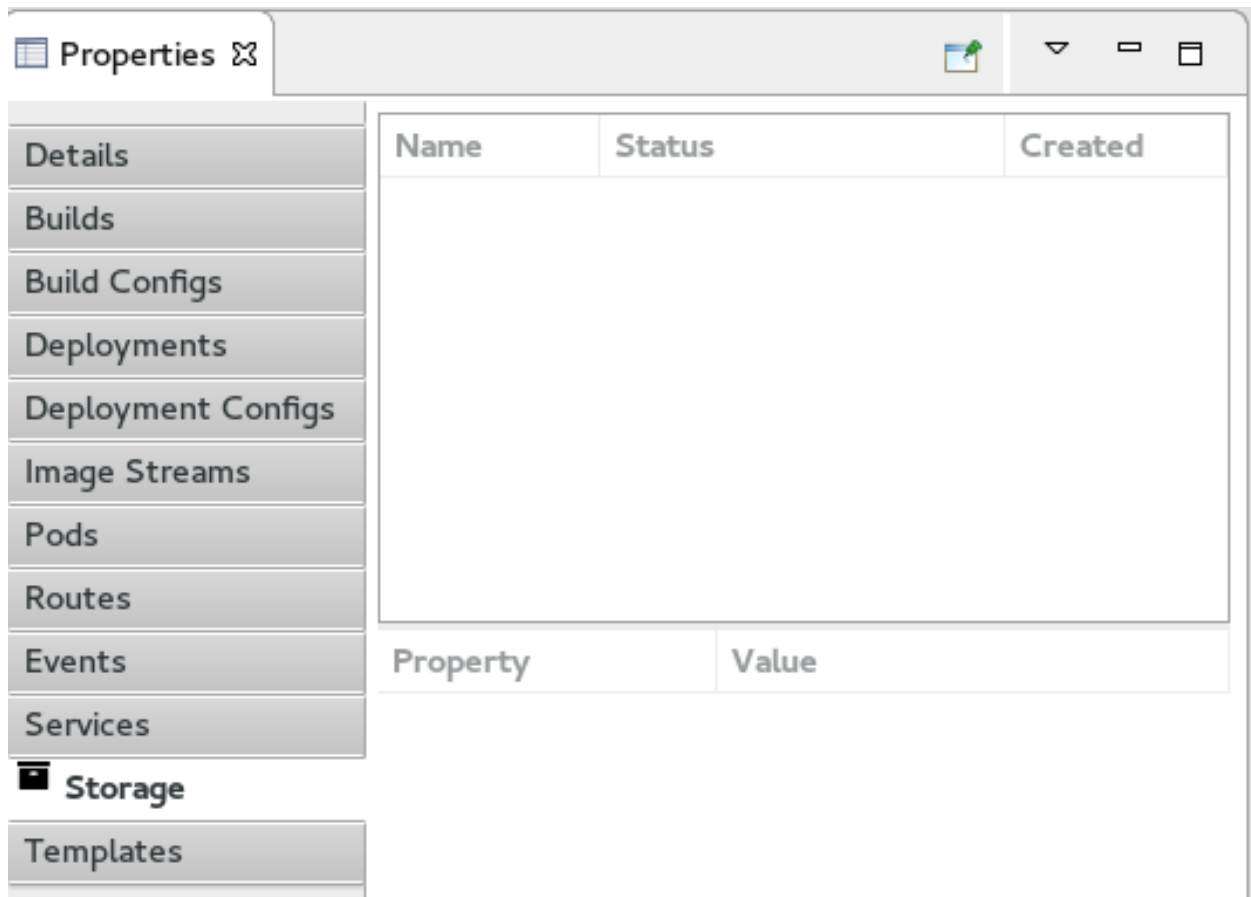
Property	Value
Count	1
First Seen	17/08/16 11 h 11 CEST
Kind	Event
Last Seen	17/08/16 11 h 11 CEST
Message	Started container with docker id 064fa
Name	nodejs-1-m0b4r
Reason	Started

You can refresh the content of the event with the **Refresh** option or open the event in the OpenShift web console with the **Show In** → **Web Console** option.

Related JIRA: [JBIDE-22817](#)

1.5.5.3. Volume claims can now be displayed

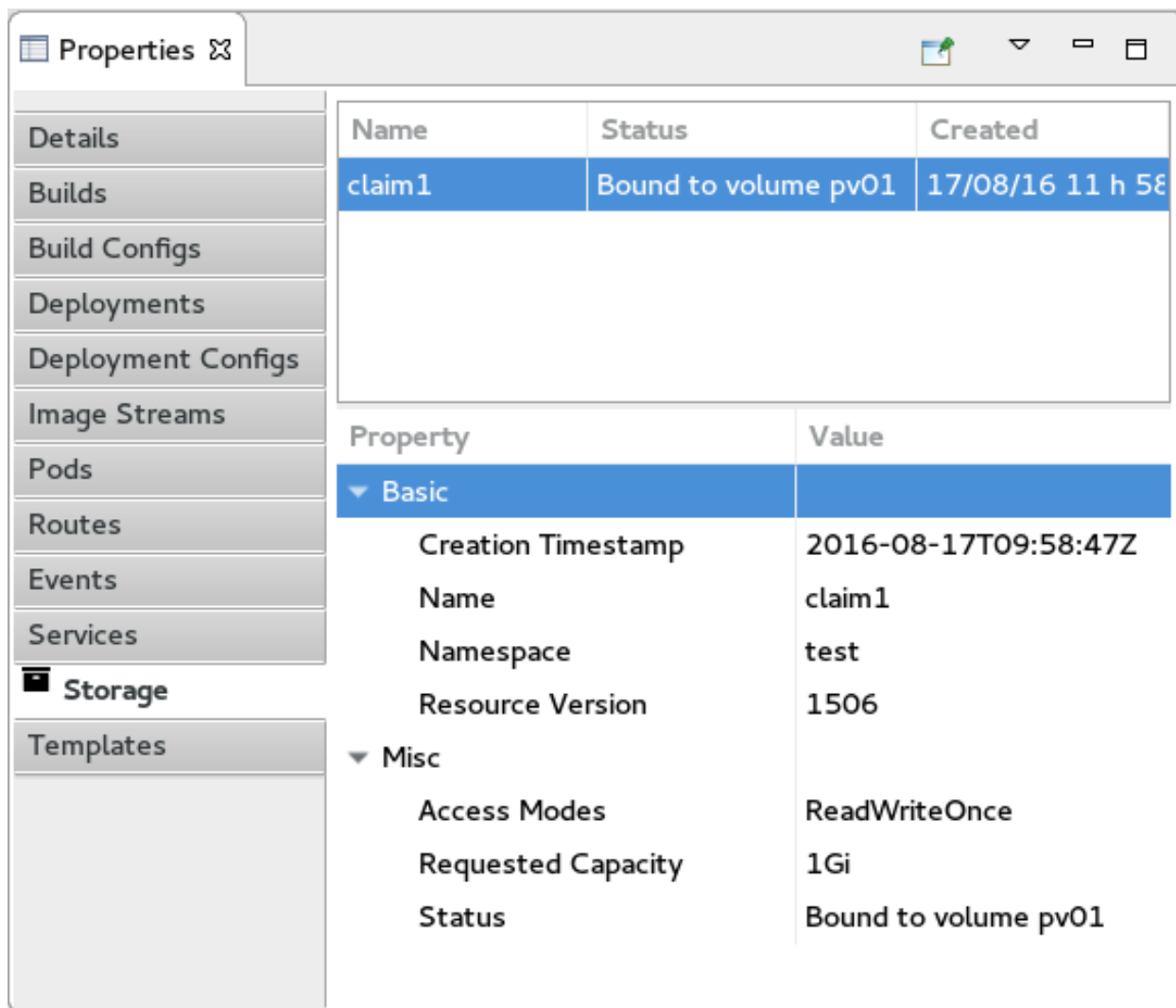
Volume claims are now displayed in the **Property** view under the **Storage** tab (available at the project level):



You can create a new volume claim using a resource file:

```
{
  "apiVersion": "v1",
  "kind": "PersistentVolumeClaim",
  "metadata": {
    "name": "claim1"
  },
  "spec": {
    "accessModes": [ "ReadWriteOnce" ],
    "resources": {
      "requests": {
        "storage": "1Gi"
      }
    }
  }
}
```

If you deploy such a resource file with the **New** → **Resource** option at the project level, the **Storage** tab is updated:



Name	Status	Created
claim1	Bound to volume pv01	17/08/16 11 h 58

Property	Value
▼ Basic	
Creation Timestamp	2016-08-17T09:58:47Z
Name	claim1
Namespace	test
Resource Version	1506
▼ Misc	
Access Modes	ReadWriteOnce
Requested Capacity	1Gi
Status	Bound to volume pv01

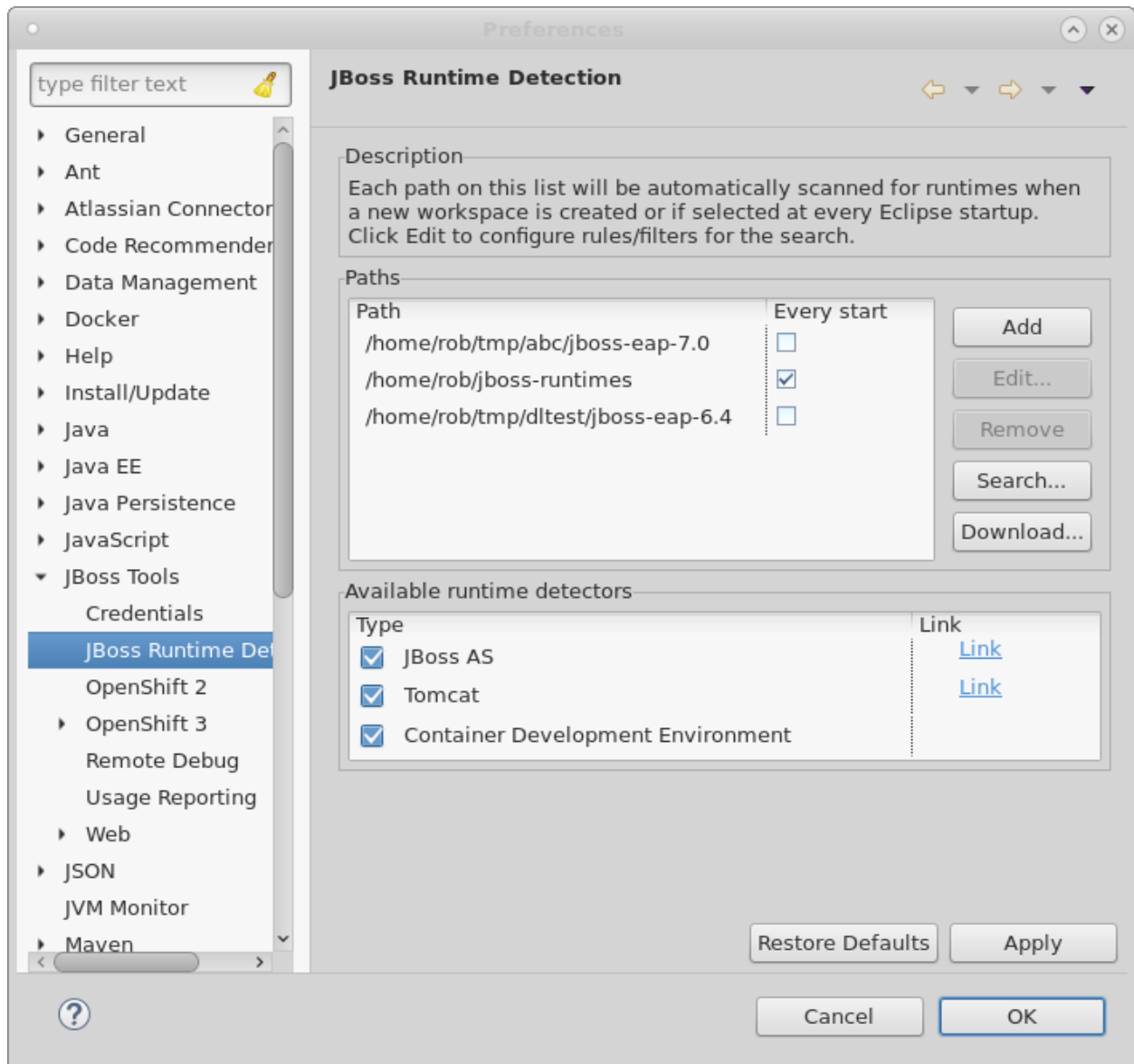
You can access/edit the content of the volume claim with the **Edit** option or open the volume claim in the OpenShift web console with the **Show In** → **Web Console** option.

Related JIRA: [JBIDE-22433](#)

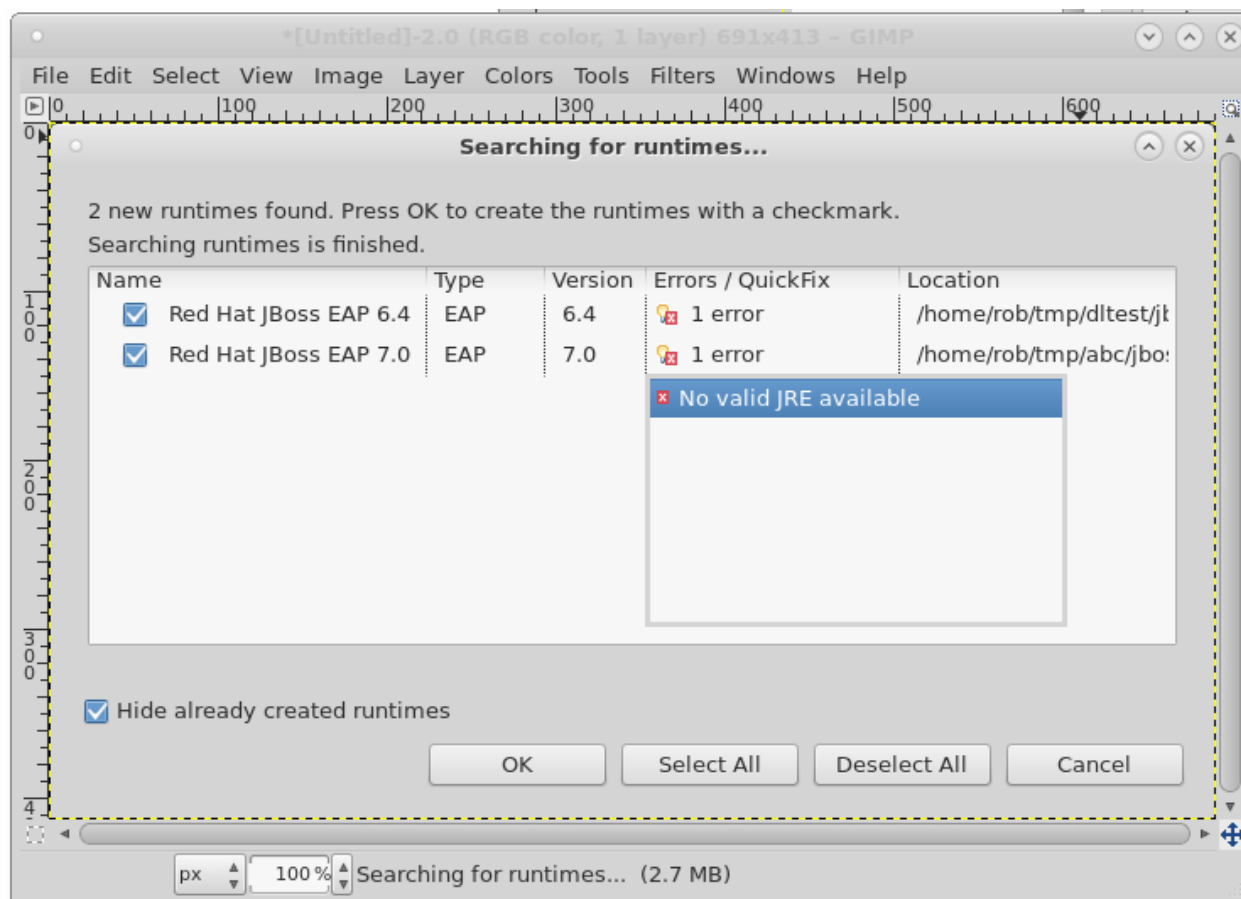
1.5.6. Server Tools

1.5.6.1. QuickFixes now available in runtime detection

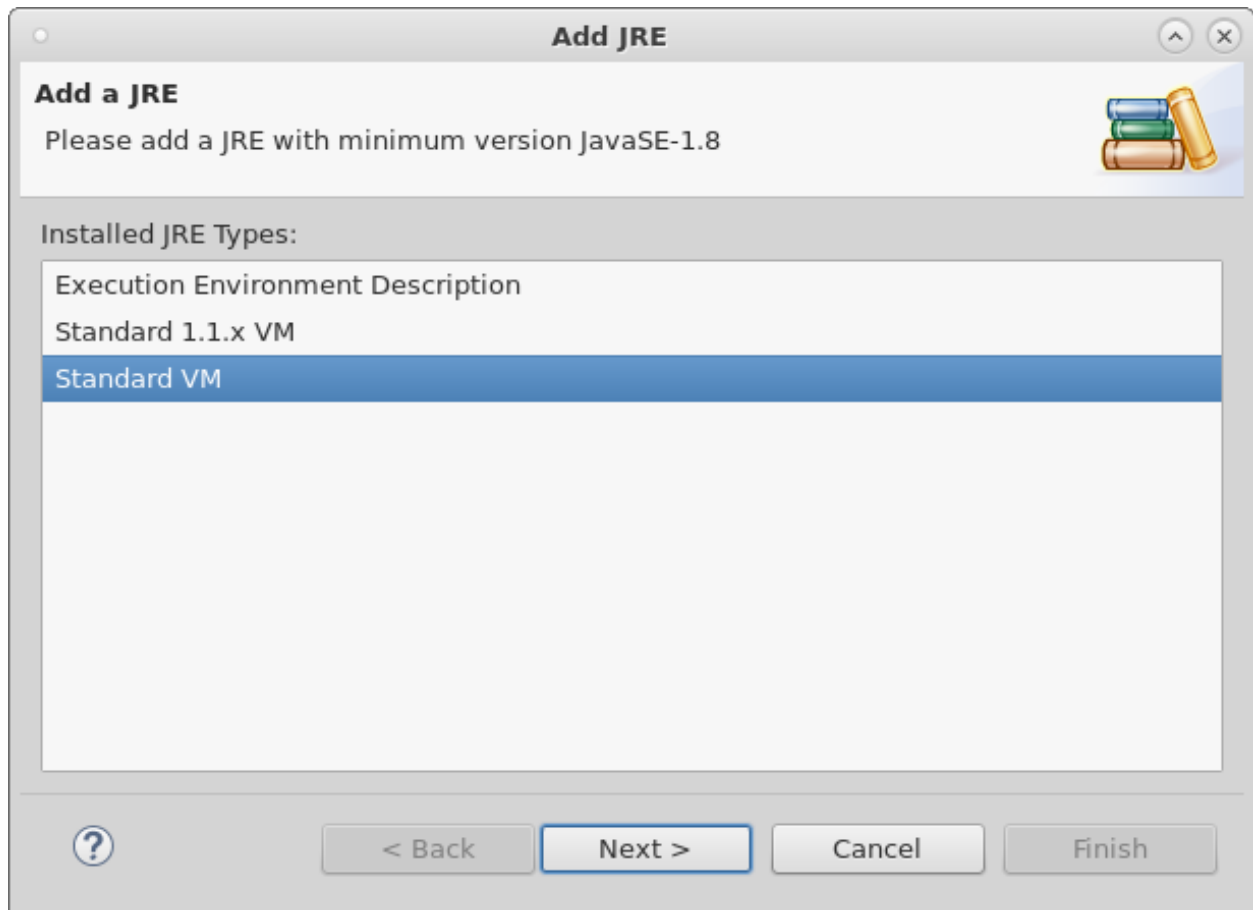
Runtime detection has been included in JBoss Tools for several releases, but it sometimes creates runtimes and server adapters with configuration errors without alerting the user. Now, the user has an opportunity to execute quick fixes before completing the creation of their runtimes and servers.



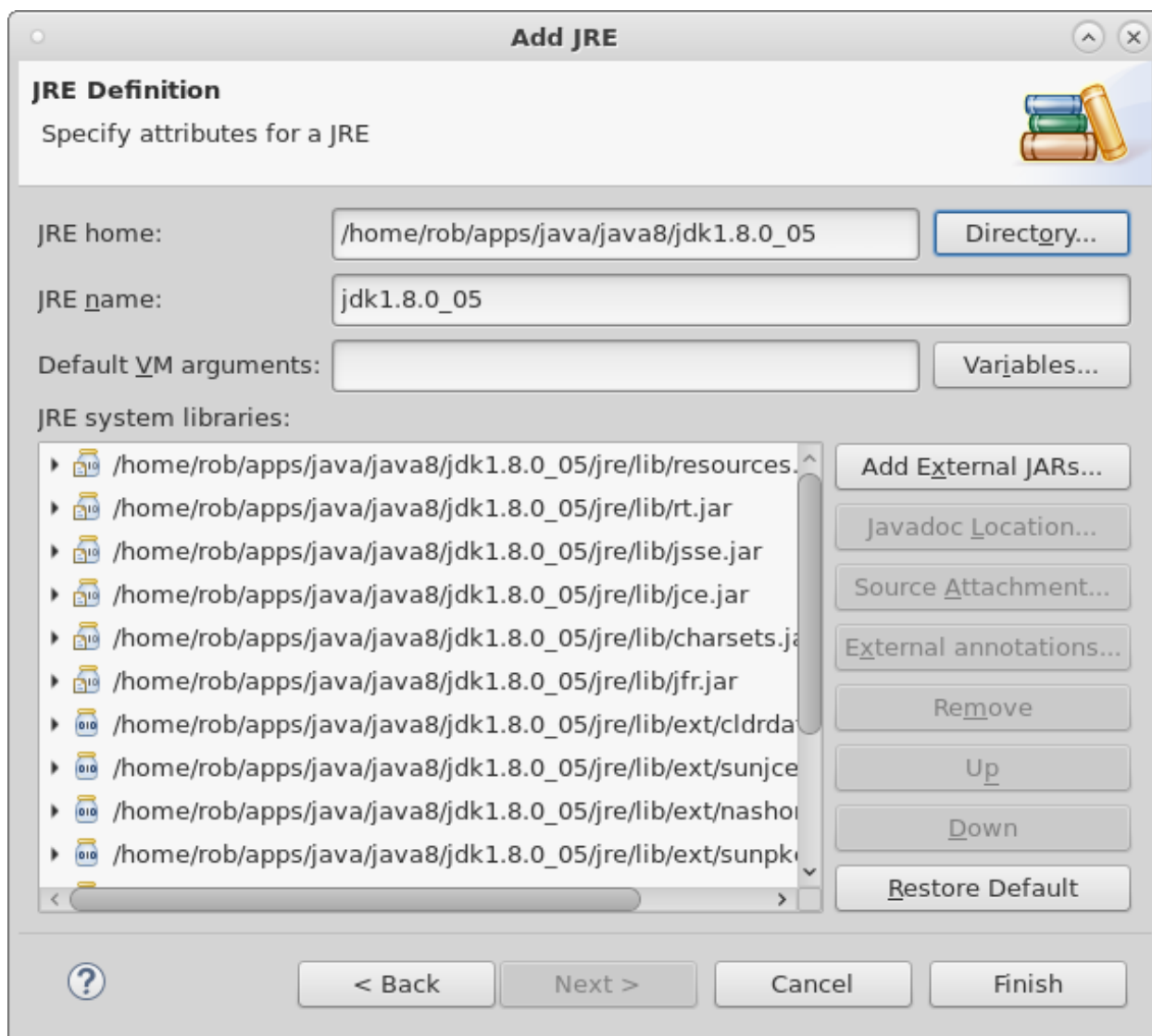
For example, at the runtime-detection preference page, the runtime-detection automatically searches the three paths for valid runtimes of any type.



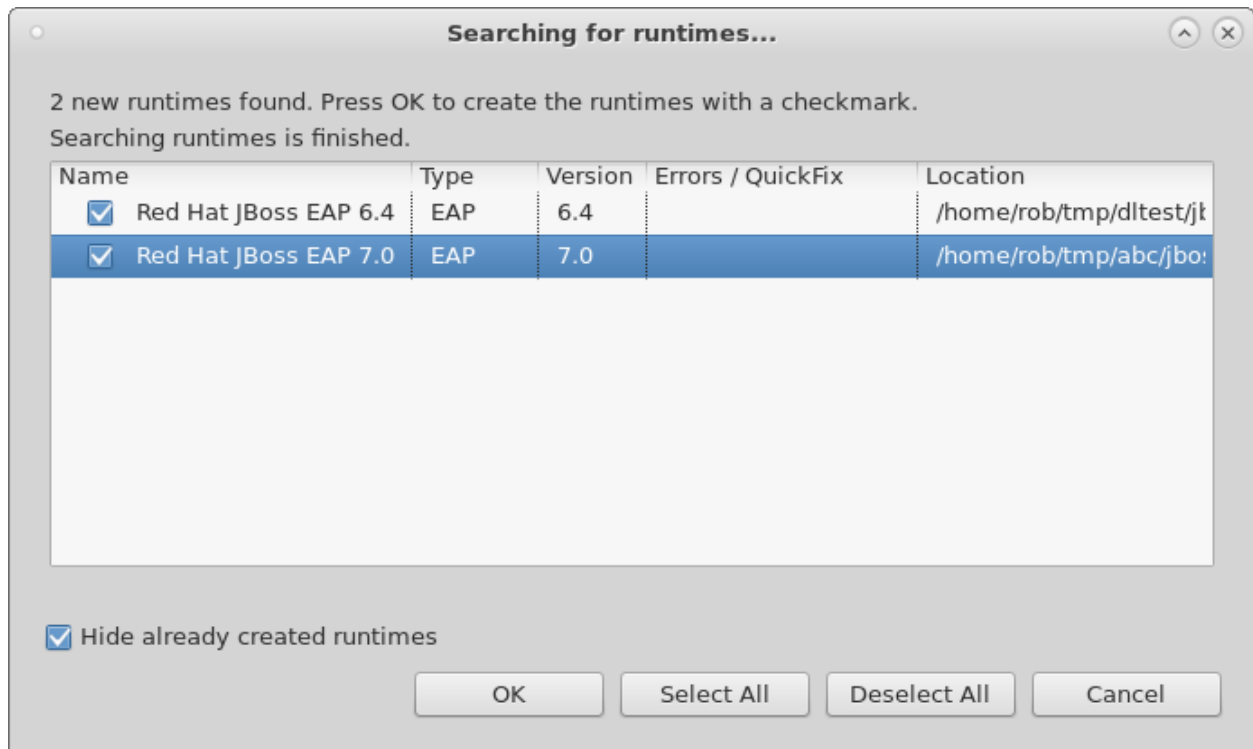
When you click **Search**, the runtime-detection's search dialog appears with the results. In this case, it has located an EAP 6.4 and an EAP 7.0 installation. However, we can see that both have errors. If we click on the error column for the discovered EAP 7.0, the error is expanded, and we see that a valid/compatible JRE is missing. To fix the issue, click this item.



A new JRE dialog box appears when you click on the problem for EAP 7, which is used to add a compatible JRE. The dialog informs us of what the restrictions are for this specific runtime. In this case, a JRE with a minimum version of Java 8 is required.



If you locate and add a Java 8 JRE, as shown above, all the errors disappear for both runtimes. In this example, the EAP 6.4 required a JRE of Java 7 or higher. The addition of the Java 8 JRE fixed this issue as well.



This dialog is intended to help preemptively discover and fix errors before using the created server adapters.

Related JIRA: [JBIDE-15189 - Add quickfixes to runtime detection](#)

1.5.7. Support for WildFly 10.1

The WildFly 10.0 Server adapter has been renamed to WildFly 10.x. It has been tested and verified to work for WildFly 10.1 installations.

Related JIRA: [JBIDE-23035](#)

1.6. ISSUES

1.6.1. Resolved Issues

To view information about resolved issues in this release of JBoss Developer Studio, see <https://issues.jboss.org/issues/?jql=> on the JBoss issue tracking system and enter the following advanced search query as one line:

```
(project in (JBDS) AND affectedVersion < 10.1.0.AM1 AND fixVersion in
(10.1.0.AM1, 10.1.0.AM2, 10.1.0.AM3, 10.1.0.GA) OR project in (JBIDE)
AND affectedVersion < 4.4.1.AM1 AND fixVersion in (4.4.1.AM1,
4.4.1.AM2, 4.4.1.AM3, 4.4.1.Final)) AND type in (Bug) AND resolution in
(Done)
```

1.7. KNOWN ISSUES

To view information about known issues in this release of JBoss Developer Studio, see <https://issues.jboss.org/issues/?jql=> on the JBoss issue tracking system and enter the following advanced search query as one line:

```
(project in (JBDS) AND (affectedVersion <= 10.1.0.GA OR affectedVersion  
is EMPTY) AND (resolution in (Unresolved) OR resolution in (Done) AND  
fixVersion > "10.1.0.GA") OR project in (JBIDE) AND (affectedVersion <=  
4.4.1.Final OR affectedVersion is EMPTY) AND (resolution in  
(Unresolved) OR resolution in (Done) AND fixVersion > "4.4.1.Final"))  
AND type in (Bug)
```

The following customer-reported known issues are highlighted:

- ✳ [JBIDE-20985](#): XML Schema validation fails when importing external schema
- ✳ [JBIDE-20983](#): cannot use oracle service name in datasource creation
- ✳ [JBIDE-19633](#): Not able to create 'non-bare' repository in JBDS 8.1.
- ✳ [JBIDE-17176](#): Unable to browse and select PortletBridge runtime libraries in JPP 6
- ✳ [JBIDE-12957](#): Xhtml files appear garbled when it's reopened in the JBDS editor
- ✳ [JBDS-3645](#): Installation of JBoss Developer Studio to a network drive fails
- ✳ [JBDS-3470](#): Toolbars + Icons unusable on UHD screens
- ✳ [JBDS-3186](#): JBDS crashes when editor is opened on RHEL 6
- ✳ [JBDS-3069](#): Ungraceful shutdown results in multiple errors on startup
- ✳ [JBDS-1080](#): Unclear text in GettingStartedGuide documentation

1.8. APPLY THIS RELEASE

JBoss Developer Studio 10.1 is available from a number of sources:

- ✳ To install JBoss Developer Studio 10.1, use the universal installer available from the Red Hat Customer Portal.
- ✳ To install JBoss Developer Studio BYOE 10.1 in Eclipse Neon, use Eclipse Marketplace, the JBoss Developer Studio update site or the update *.zip* file available from the Red Hat Customer Portal.

In all cases, for more information see the Red Hat JBoss Developer Studio Installation Guide at the [Red Hat JBoss Developer Studio Documentation](#) page.