



Red Hat Developer Studio 12.0

Release Notes and Known Issues

Highlighted features in 12.0

Red Hat Developer Studio 12.0 Release Notes and Known Issues

Highlighted features in 12.0

Supriya Takkhi
sbharadw@redhat.com

Legal Notice

Copyright © 2018 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 Developer Studio 12.0.

Table of Contents

| | |
|--|-----------|
| CHAPTER 1. INTRODUCTION TO RED HAT DEVELOPER STUDIO | 3 |
| 1.1. ABOUT RED HAT DEVELOPER STUDIO | 3 |
| 1.2. USE CASES OF DEVELOPER STUDIO | 3 |
| 1.2.1. Web Applications | 3 |
| 1.2.2. Web Applications Optimized for Mobile Devices | 4 |
| 1.2.3. Applications for Cloud Deployment | 4 |
| 1.3. ABOUT THIS RELEASE | 4 |
| 1.4. ECLIPSE AUTOMATED REPORTING INTERFACE (AERI) | 4 |
| CHAPTER 2. RELEASE NOTES | 5 |
| 2.1. FORGE TOOLS | 5 |
| 2.1.1. Forge Runtime updated to 3.9.0.Final | 5 |
| 2.2. FUSE TOOLING | 5 |
| 2.2.1. Camel Rest DSL from WSDL Wizard | 5 |
| 2.2.2. Camel Editor REST Tab | 8 |
| 2.2.3. Camel URI Completion with XML DSL | 9 |
| 2.3. MAVEN | 9 |
| 2.3.1. Maven Support Updated to M2E 1.9.1 | 9 |
| 2.4. OPENSIFT | 10 |
| 2.4.1. Enhanced Spring Boot Support for Server Adapter | 10 |
| 2.5. SERVER TOOLS | 10 |
| 2.5.1. Wildfly 13 Server Adapter | 10 |
| 2.6. WEB SERVICES TOOLS | 10 |
| 2.6.1. JAX-RS 2.1 Support | 10 |
| 2.6.2. Server Side Events | 10 |
| CHAPTER 3. ISSUES | 11 |
| 3.1. RESOLVED ISSUES FOR RED HAT DEVELOPER STUDIO | 11 |
| 3.2. KNOWN ISSUES FOR RED HAT DEVELOPER STUDIO | 11 |
| 3.3. KNOWN ISSUES FOR RED HAT FUSE | 11 |
| CHAPTER 4. ADDITIONAL RESOURCES | 12 |

CHAPTER 1. INTRODUCTION TO RED HAT DEVELOPER STUDIO

1.1. ABOUT RED HAT DEVELOPER STUDIO

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).

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. Developer Studio can also 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
- Incorporate Hibernate, CDI, JAX-RS, JSF, and other popular APIs into applications with simplicity using the tool-driven interface
- Deploy applications to JBoss runtime servers and the cloud using JBoss Server Tools and OpenShift Tools

Developer Studio is built around Eclipse and packaged with all the necessary dependencies and third-party plug-ins for simplified installing. For developers already running Eclipse, Developer Studio can also be installed through Eclipse Marketplace. Installing Developer Studio in an existing Eclipse installation is referred to as BYOE (Bring Your Own Eclipse).

1.2. USE CASES OF DEVELOPER STUDIO

Developer Studio assists Java EE developers by integrating JBoss technology and APIs in a single development environment. Here are a few ways that Developer Studio helps make development easier:

1.2.1. Web Applications

Red Hat Central provides wizards that generate skeletons 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. Developer Studio also offers project file templates in a range of popular programming languages, including HTML, XHTML, and JSF.

Palettes in Developer Studio give access to the core elements of the JSF and RichFaces 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. 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 avoid needing to manually refresh. You can 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.

1.2.3. 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

Red Hat Developer Studio 12.0 is an update of Red Hat Developer Studio 11.3 and it has the following features:

- It includes Eclipse Photon (4.8.0).
- It requires a minimum of Java 8 to run.
- It introduces new features, which are outlined in the Release Notes section.
- It contains new features for the existing tools.
- It resolves issues identified in earlier versions of Developer Studio.

For more information about operating systems, chip architectures and Java developer kits supported by this release, see [Supported Configurations and Components](#) page on the Red Hat Customer Portal.

1.4. ECLIPSE AUTOMATED REPORTING INTERFACE (AERI)

To contribute to JBoss Tools, we recommend you to enable the Eclipse Automated Reporting Interface (AERI) in JBoss Tools. To read about configuring error reporting in JBoss Tools, see: <http://tools.jboss.org/usage/#error-reporting>.

CHAPTER 2. RELEASE NOTES

2.1. FORGE TOOLS

2.1.1. Forge Runtime updated to 3.9.0.Final

The included Forge runtime is now 3.9.0.Final. Read the official announcement [here](#).

2.2. FUSE TOOLING

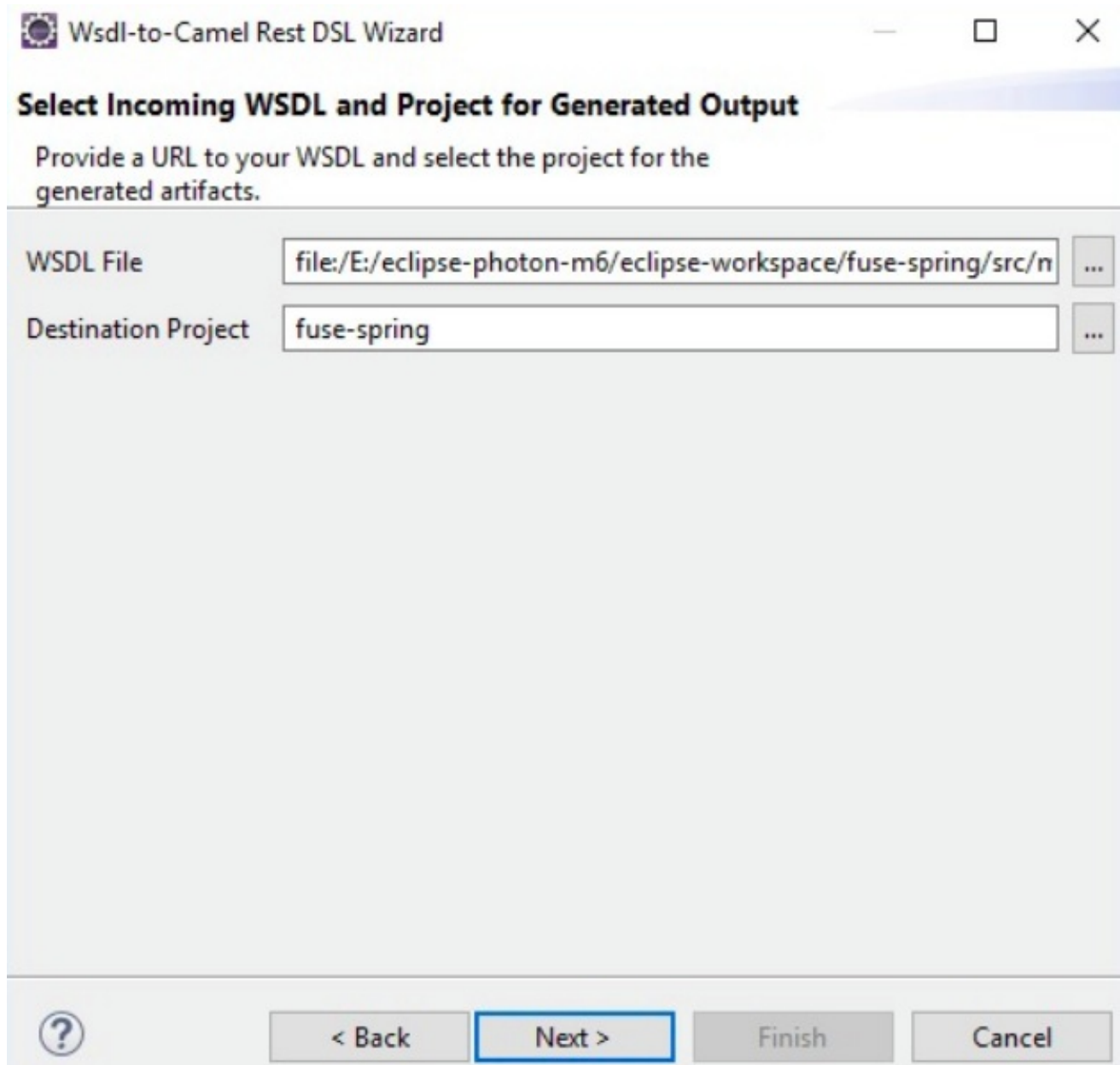
2.2.1. Camel Rest DSL from WSDL Wizard

The new **Camel Rest DSL from WSDL** wizard wraps the `wsdl2rest` tool that is now included with the Fuse 7 distribution. This tool takes a WSDL file for a SOAP-based (JAX-WS) web service and generates a combination of CXF-generated code and a Camel REST DSL route to make it accessible using REST operations.

To start, you need an existing Fuse Integration project in your workspace and access to the WSDL for the SOAP service. Then use the **File > New > Other** menu and select **Red Hat Fuse > Camel Rest DSL** from the **WSDL** wizard.

On the first page of the wizard, select your WSDL and the Fuse Integration project in which you want to generate the Java code and Camel configuration.

Figure 2.1. Selecting the WSDL and Fuse Integration Project



You can customize the Java folder path for your generated classes, the folder for the generated Camel file, and any customization for the SOAP service address and destination REST service address.

Figure 2.2. Specifying the Advanced Options for wsdl2rest Processing

WsdI-to-Camel Rest DSL Wizard

Specify Advanced Options for wsdl2rest Processing

⚠ Path as specified does not exist in project and will be created.

Destination Java Folder ...

Destination Camel Folder ...

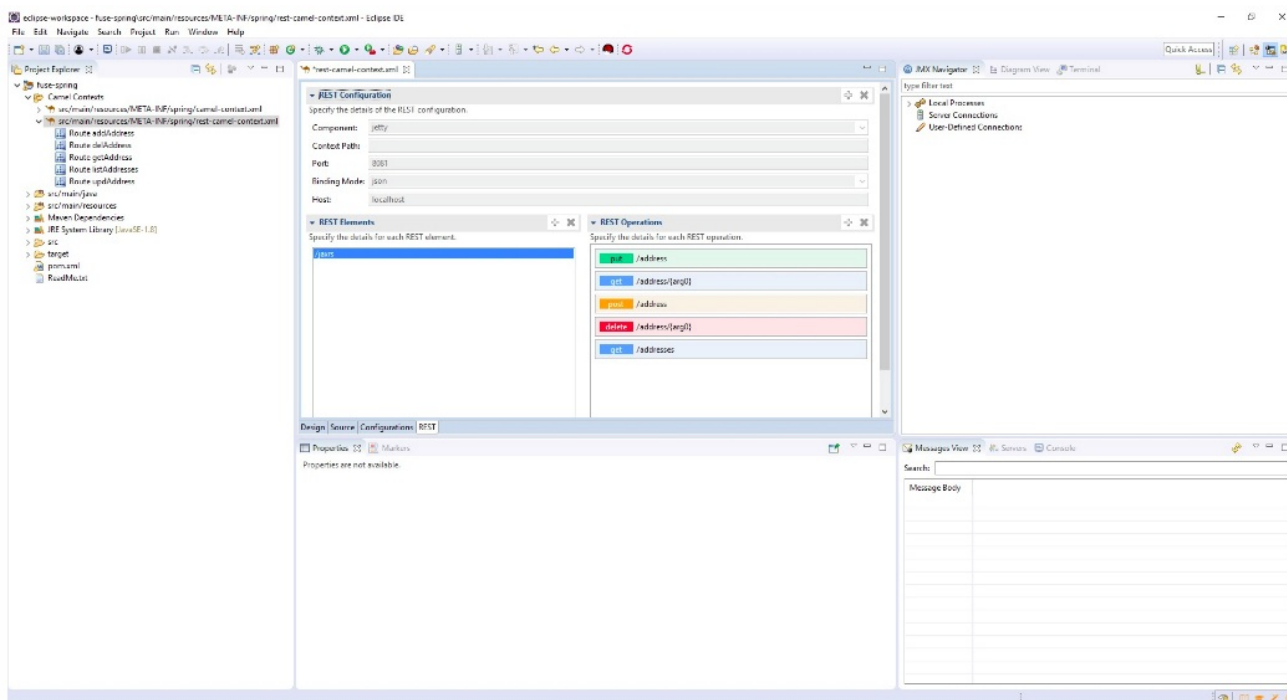
Target Service Address

Target REST Service Address

? < Back Next > Finish Cancel

Click **Finish** to generate the new Camel configuration and the associated Java code in your project. The wizard determines whether your project is Blueprint, Spring, or Spring Boot based, and it creates the corresponding artifacts without requiring any additional input. When the wizard is completed, you can open your new Camel file in the Fuse Tooling Route Editor to view its content.

Figure 2.3. Viewing the Camel File Content

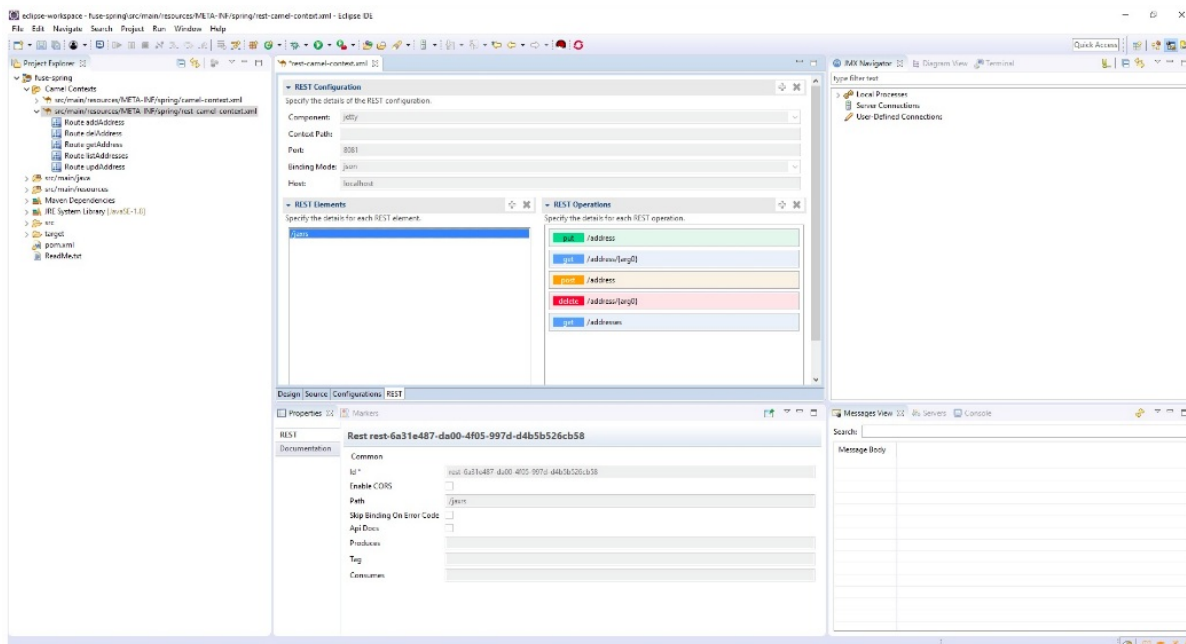


2.2.2. Camel Editor REST Tab

The Fuse Tooling Route Editor provides a new **REST** tab. For this release, the contents of this tab are read-only and include the following information:

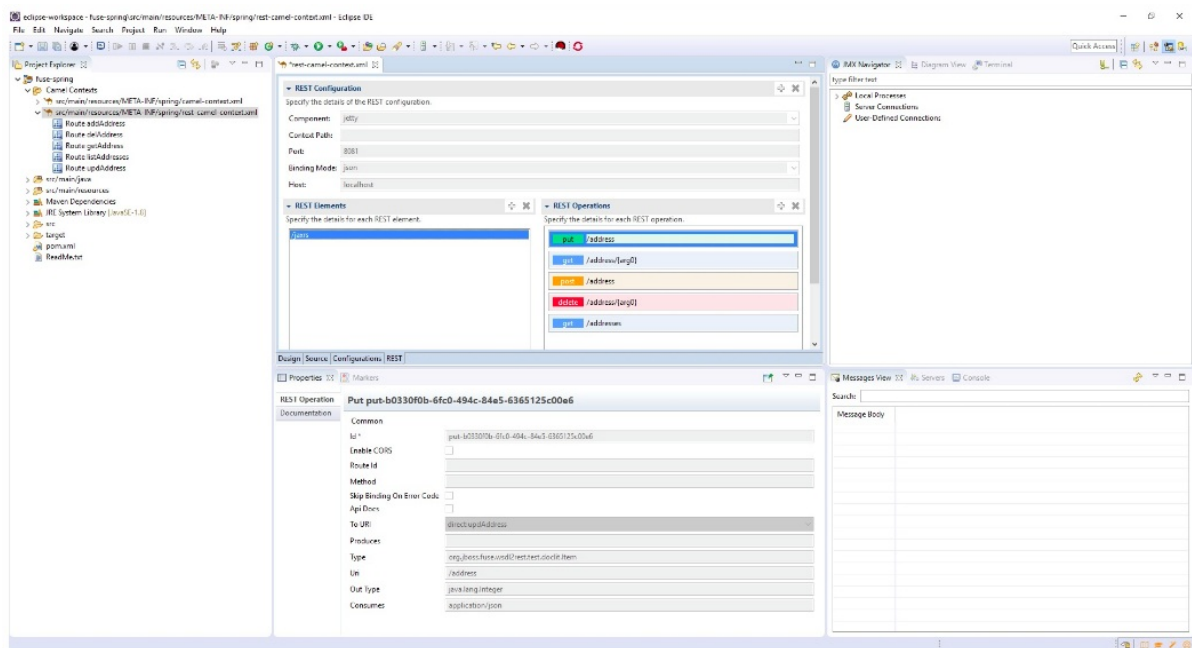
- Details for the REST Configuration element including the component (jetty, netty, servlet, etc.), the context path, the port, binding mode (JSON, XML, etc.), and host. There is only one REST Configuration element.
- A list of REST elements that collect REST operations. A configuration can have more than one REST element. Each REST element has an associated property page that displays additional details such as the path and the data it consumes or produces.

Figure 2.4. List of REST Elements that Collect REST Operations



- A list of REST operations for the selected REST element. Each of the operations has an associated property page that provides details such as the URI and output type.

Figure 2.5. List of REST Operations for the Selected REST Element



For this release, the REST tab is read-only. If you want to edit the REST DSL, use the **Route Editor Source** tab. When you make changes and save them in the **Source** tab, the **REST** tab refreshes to show your updates.

2.2.3. Camel URI Completion with XML DSL

It was possible to have Camel URI completion with XML DSL in the **Source** tab of the Camel Route editor by installing the Language Support for Apache Camel in your IDE.

This feature is now installed by default with Fuse Tooling. To view the feature in action, refer to <https://tools.stage.jboss.org/documentation/whatsnew/fusetools/images/completionSourceEditor.gif>.

Depending on your development preference, you can now use the **Properties** view with UI help to configure Camel components or use the source editor and benefit from completion features.

2.3. MAVEN

2.3.1. Maven Support Updated to M2E 1.9.1

The Maven support is based on Eclipse M2E 1.9.1, bringing the following features:

- **Advanced classpath isolation:** Due to Eclipse Photon, two new classpaths are now available: the main and the test classpath. The main classes will now no longer see the test classes and dependencies.
- **Embedded Maven runtime:** The embedded Maven runtime is now based on Apache Maven 3.5.3.
- **Archetype catalog management:** It is now possible to disable an archetype catalog.

- Java 9/10 support: Support for Java 9/10 has been improved: bugs fixes, better handling of module path.

2.4. OPENSIFT

2.4.1. Enhanced Spring Boot Support for Server Adapter

Spring Boot runtime was already supported by the OpenShift server adapter. One of its major limitation has now been fixed. The limitation was that files and resources were synchronized between the local workstation and the remote pod(s) only for the main project. If your Spring Boot application had dependencies that were present in the local workspace, any changes to a file or resource of one of these dependencies was not handled.

Related JIRA: [JBIDE-25504](#)

2.5. SERVER TOOLS

2.5.1. Wildfly 13 Server Adapter

A server adapter has been added to work with Wildfly 13. It adds support for Servlet 4.0.

Related JIRA: [JBIDE-26071](#)

2.6. WEB SERVICES TOOLS

2.6.1. JAX-RS 2.1 Support

JAX-RS 2.1 is part of JavaEE8 and JBoss Tools now provides you with support for this update of the specification.

2.6.2. Server Side Events

JAX-RS 2.1 brought support for server side events. The Sse and SseEventSink resources can now be injected into the method arguments, thanks to the `@Context` annotation.

Related JIRA: [JBIDE-26040](#)

CHAPTER 3. ISSUES

3.1. RESOLVED ISSUES FOR RED HAT DEVELOPER STUDIO

To view information about resolved issues in this release of Developer Studio, see the [Resolved Issues](#).

3.2. KNOWN ISSUES FOR RED HAT DEVELOPER STUDIO

To view information about known issues in this release of Developer Studio, see the [Known Issues](#).

The following known issues are highlighted:

- [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 Developer Studio to a network drive fails
- [JBDS-3470](#): Toolbars + Icons unusable on UHD screens
- [JBDS-3069](#): Ungraceful shutdown results in multiple errors on startup
- [JBDS-4442](#): Central page does not work on Fedora 26 if package webkitgtk3 is not installed
- [JBIDE-25146](#): Eclipse annotation processing not enabled by default can result in errors

3.3. KNOWN ISSUES FOR RED HAT FUSE

Following is a list of issues for the DevStudio and Fuse integration:

- [FUSETOOLS-2859](#): For a project that contains a data transformation file, if you change the Camel version from a version earlier than 2.20 to version 2.20 or later, you must open the data transformation file in the Data Transformation editor to complete the Camel version change. Otherwise, the project creates an error at runtime.

CHAPTER 4. ADDITIONAL RESOURCES

Developer Studio 12.0 is available from a number of sources:

- To install Developer Studio 12.0, use the universal installer available from the [Red Hat Customer Portal](#) or from the [Red Hat Developer Program](#).
- To install Developer Studio BYOE 12.0 in Eclipse Oxygen, use Eclipse Marketplace, the 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 Developer Studio Installation Guide at the [Developer Studio Documentation](#) page.