CDI Tools Reference Guide

Version: 3.3.0.M5

1. Introduction to CDI tooling	1
2. Creating a CDI Web Project	3
3. Adding CDI Support to a project	17
4. Wizards and Dialogs	19
4.1. Wizards	19
4.2. Dialogs	20
4.2.1. Open CDI Named Bean	20
5. Validation	23
6. Hyperlink navigation	27
A. Revision History	31

Introduction to CDI tooling

Context and Dependency Injection (CDI) tooling allows you to easily add and work with the CDI programming model for your web-based applications. This guide demonstrates how to add CDI to existing projects and create new projects with the CDI wizard. For information on the specifics of CDI outside of the tooling, see *http://jcp.org/en/jsr/detail?id=299*.

Creating a CDI Web Project

From the workbench go to $\textbf{File} \rightarrow \textbf{New} \rightarrow \textbf{Other}.$



Figure 2.1. Selecting the CDI Web Project wizard

The first screen of the wizard will ask you to define the attributes according to the options outlined in *Table 2.1, "New CDI Web Project*". Click the **Next** button after defining the attributes in order to continue customizing your project or click the **Finish** button to accept all other defaults and create your application.

CDI Web Project	
Create a standalone CDI Web project or add it to a new or existing Enterpr Application.	rise
Project na <u>m</u> e: testCDI	
Project location	
Use default location	Require
	DIOWSE
JBoss EAP 6.0 Runtime	ew <u>R</u> untime
Dynamic web module version	
3.0	\$
Configuration	
Dynamic Web Project with CDI (Context and Dependency Injectio	Mod <u>i</u> fy
Configures a Dynamic Web application to use Context and Dependency Inj	ection (CDI)
EAR membership	
EAR project name: testCDIEAR	lew <u>P</u> roject
Working sets	
Add project to working sets	
Working sets:	S <u>e</u> lect
? < Back Next > Cancel	Finish

Figure 2.2. Main project creation screen

Field	Mandatory	Instruction	Description
Project name	yes	Enter a project name.	The project name can be any name you define.
Project location	yes	The Use default location checkbox will be selected automatically to define the project location as the Eclipse workspace. However you can define a custom path in the Location field be deselecting the Use default location checkbox.	The default location corresponds to the Eclipse workspace.
Target runtime	no	Select a pre-configured runtime from the available options or configure a new runtime environment.	The target runtime defines the server to which the application will be deployed.
Dynamic web module version	yes	Select the required web module version.	This option adds support for the Java Servlet API with module versions corresponding to JEE or J2EE levels as listed in <i>Table 2.2,</i> <i>"New CDI Web Project - Dynamic</i> <i>web module version"</i> .
Configurati	omjes	Select the project configuration from the available options.	The project can be based on either a custom or a set of pre-defined configurations as described in <i>Table 2.3, "New CDI</i> <i>Web Project - Configuration"</i> .
EAR membersh	no ip	Add the project to an existing EAR project.	The project can be added to an existing EAR project by selecting the checkbox. Once checked, a new EAR project can be defined by clicking the New Project button.
Working sets	no	Add the project to an existing working set.	A working set provides the ability to group projects or project attributes in a customized way to improve access. A new working set can be defined once the Select button has been clicked.

Table 2.1. New CDI Web Project

Option	Description
3.0	This web module version corresponds to the JEE 6 implementation.
2.5	This web module version corresponds to the JEE 5 implementation.
2.4	This web module version corresponds to the J2EE 1.4 implementation.
2.3	This web module version corresponds to the J2EE 1.3 implementation.
2.2	This web module version corresponds to the J2EE 1.2 implementation.

Table 2.2. New CDI Web Project - Dynamic web module version

Table 2.3. New CDI Web Project - Configuration

Option	Description
Dynamic Web Project with CDI	The default configuration option. Choosing this configuration will create a Dynamic Web Project with the CDI facet enabled.
Default configuration for the runtime specified in Target runtime	Depending on the runtime configuration, you may have to manually add CDI facet support through the Modify button.
JavaServer Faces v2.0 Project	Configures a project to use JSF v2.0. You will need to manually add CDI support to this configuration through the Modify button.
Minimal Configuration	The minimum required facets are installed. You will need to manually add CDI support to this configuration through the Modify button.
<custom></custom>	Total manual configuration is required through the Modify button. Ensure you enable the CDI facet for CDI Web Project creation to succeed.

The Java configuration screen allows you to define **Source folders on the build path** and the **Default output folder**. The defaults for each are recommended, however you have the option to change them as you see fit.

Java				
Configure project for l	building a Java app	lication.		
<u>S</u> ource folders on buil	d path:			_
<pre> @# src </pre> Default output folder:				Add Folder
build/classes				
?	< Back	Next >	Cancel	Finish

Figure 2.3. Java configuration screen

The **Web Module** configuration screen allows you to define the **Context root** and the **Content directory** to be created, the option to **Generate web.xml deployment descriptor** is selected by default. The defaults for each are recommended, however you have the option to change them as you see fit.

Web Module Configure web mod	lule settings.	
Context <u>r</u> oot:	testCDI	
Content <u>d</u> irectory:	WebContent	
☑ <u>G</u> enerate web.x	ml deployment descriptor	
?	< Back Next > Cano	cel Finish

Figure 2.4. Web Module configuration screen

The **Context and Dependency Injection (CDI) Facet** settings configuration screen allows you to **Generate beans.xml file**, selected by default. The beans.xml file is a bean archive descriptor that allows you to enable CDI services for a bean archive. It is recommended that you leave the option selected.

Context and Depend Configure Context and	dency Injection (CDI) Facet d Dependency Injection (CDI) Fa	cet Settings	
Generate beans.xml f	ile: 🗹		
?	< Back Next >	Cancel	Finish

Figure 2.5. Web Module configuration screen

The final screen of the wizard pertains to **JSF Capabilities** configuration.

JSF Capabilities			
Add JSF capabilities to this	s Web Project		
JSF Implementation Librar	у		
Type: Library Provided b	y Target Runtime	\$	
The targeted runtime is al option will configure the p	ble to provide the library required by this facet. Sele project to use that library.	cting this	
✓ Configure <u>J</u> SF servlet in	n deployment descriptor		
JSF <u>C</u> onfiguration File:	/WEB-INF/faces-config.xml		
JSF <u>S</u> ervlet Name:	Faces Servlet		
JSF Servlet C <u>l</u> ass Name:	javax.faces.webapp.FacesServlet		
<u>U</u> RL Mapping Patterns:	*.seam	<u>A</u> dd <u>R</u> emove	
?	< Back Next > Cancel	Finish	

Figure 2.6. Web Module configuration screen

Field	Mandatory	Instruction	Description
JSF Implementa Library Type	yes ation	Select an implementation library from the list provided.	This option adds support for the JSF implementation library, required for the project to function correctly. For a description of available options see: Table 2.5, "JSF Capabilities - JSF Implementation Library".
Configure JSF servlet in deploymen descriptor	no t	Selected by default, decide whether to configure the deployment descriptor for the JSF servlet or not.	This option, if selected, opens up the ability to configure various deployment descriptor fields. For a description of fields see: <i>Table 2.6, "JSF Capabilities - JSF</i> <i>Servlet Configuration"</i> .

Table 2.4. JSF Capabilities

Table 2.5. JSF Capabilities - JSF Implementation Library

Option	Description
Library Provided by Target Runtime	If you selected a target runtime on the first screen, this option will configure the project to use the required facet library that is provided by the target runtime.
User Library	Selecting the User Library option allows you to specify one or more custom user libraries to be used. At least one must be selected for the wizard to complete successfully.
Disable Library Configuration	If you choose to disable library configuration you will have to configure the project classpath at a later time, by alternate means, before running the project.

Table 2.6. JSF Capabilities - JSF Servlet Configuration

Field	Mandatory	Instruction	Description
JSF Configurati File	Yes on	Enter the location of the JSF configuration file (the default is usually correct).	You must specify where the JSF configuration file is located.
JSF Servlet Name	Yes	Specify a name for your JSF servlet	The name of the servlet.
JSF Servlet Class Name	No	Specify a name for your servlet class.	Naming your servlet class will allow for easy reference to it when you are coding for your project in the future.

Field	Mandatory	Instruction	Description
URL	No	A default mapping pattern is	This option allows for the creation
Mapping		provided. You are able to remove	of constant, readable URLs,
Patters		and add patterns as you wish.	based on patterns, that map to
			your project.

Click the **Finish** button to create your CDI Web Project.

Adding CDI Support to a project

CDI support can be added either to an existing project or during project creation.

To add CDI support to an existing project, select the project, and right-click to bring up the context menu. From the menu select **Configure** \rightarrow **Add CDI support**. The following preferences page will now display:

E Pr	operties for cdiwar 🛛 🔀
	CDI (Contexton) Settings 🖉 🖓 🚽
Resource Builders	CDI support: 🗹
 CDI (Context and Depe Deployment Assembly Expression Language V Hibernate Settings Java Build Path Java Code Style Java Compiler Java Editor 	
→ JavaScript	Restore <u>D</u> efaults <u>Apply</u>
?	Cancel OK

Figure 3.1. CDI project settings

You can also add CDI support to a project during project creation. To do so, click the **Modify** button beside the selected **Configuration** on the first screen of the project wizard.

3.0		~
Configuration		
Default Configuration for JBoss EAP 6.0 Runtime		Modify
A good starting point for working with JBoss EAP 6.0 Runtime runtime. Addition installed to add new functionality to the project.	ional facets	s can later be
FAR membership		

Figure 3.2. Modifying project configuration

From the **Project Facets** screen, add CDI to the project by selecting the checkbox beside the facet **CDI (Contexts and Dependency Injection)**. Click the **OK** button to accept the configuration change and return to the project creation wizard.

Configuration: <pre><custom></custom></pre>		\$	s
Project Facet	Version	Details Runtim	nes
 Axis2 Web Services BIRT Charting Runtime Component BIRT Reporting Runtime Component 	3.7.2	CDI (Contex Injection) 1	xts a .0
E CDI (Contexts and Dependency Injection)	1.0	Enables the pro	ject i jectio
🗌 🗎 CXF 2.x Web Services	1.0	OR	
🗹 👼 Dynamic Web Module	3.0	- Requires the f	ollow
🖌 🛃 Java	1.6		01101
🗌 🗎 JavaScript	1.0	Java 1.5 ol	r nev
🗌 🗎 JavaServer Faces	2.0	Requires all of	the
🗌 🗎 JAX-RS (REST Web Services)	1.1	🗸 🛛 🗔 Dynamic W	/eb N
□ ↔ JAXB	2.1	• Java 1.5 or	r nev
🗌 🗎 JBoss Maven Integration	1.0		
🕨 🔲 🗎 JBoss Portlets			
🗌 🏄 JBoss Web Services Core	3.0	-	
□ ↔ JPA	2.0	-	
🗌 🖹 Seam	2.3	▼	

Figure 3.3. Adding CDI facet support

Wizards and Dialogs

4.1. Wizards

A set of wizards are available for creating a range of basic CDI artifacts. The wizards available are for the following types:

- Annotation Literal
- Bean
- Decorator
- Interceptor
- Interceptor Binding Annotation
- Qualifier Annotation
- Scope Annotation
- Stereotype Annotation

Select a wizard	
Wizards:	
type filter text	4
	^
Annotation Literal	
🕝 Bean	
🕖 CDI Web Project	=
G Decorator	
File beans.xml	
G Interceptor	
Interceptor Binding Annotation	
Qualifier Annotation	
Scope Annotation	
Stereotype Annotation	~
Or Cancel	Finish

Figure 4.1. Artifact wizard options

4.2. Dialogs

4.2.1. Open CDI Named Bean

The **Open CDI Named Bean** dialog allows you to open the resource which declares a specified expression language (EL) bean name.

*an		4
Matching items:		
😴 cdiBeanTestFoo3 - cdi.te	st.CDIBeanTest - /CDITest1/src	
÷	Workspace matches	
🕝 another - org.jboss.jsr29	9.tck.tests.jbt.refactoring.Another	One - /tck/JavaSour
ChangePasswordAction	org.jboss.seam.security.managem	ent.action.ChangeP
ChangePasswordAction	org.jboss.seam.security.managem	ent.action.ChangeP
contextInstance - org.jb	oss.seam.faces.context.RenderScop	pedContext.getCon
contextInstance - org.jb	oss.seam.faces.context.RenderScop	pedContext.getCon
getAnotherFunnelWeav	e r5 - org.jboss.jsr299.tck.tests.jbt.va	alidation.producers
AndledException - org.	boss.seam.exception.control.Excep	otionHandlerDispat
AndledException - org.	boss.seam.exception.control.Excep	otionHandlerDispat
Contract MyRean - cdi test MyRea	n - /CDITest1/src	

Figure 4.2. Open CDI Named Bean dialog

This dialog is available from the Java EE perspective's tool bar.



Figure 4.3. Toolbar icon

Validation

CDI validation covers nearly all of the rules declared in the JSR-299 specification. For every type of validation rule, you can set if it should be interpreted as an error, warning or just ignored. This is achieved by navigating to **Window** \rightarrow **Preferences** \rightarrow **JBoss Tools** \rightarrow **CDI** \rightarrow **CDI Validator**.



Figure 5.1. Validation settings

The CDI validator can check JSP, XHTML, beans.xml and java source files. Some validation issues can be fixed through the Quick Fixes menu (Ctrl+1).



Figure 5.2. Quick Fixes

Hyperlink navigation

Hyperlinks (OpenOn integration) allows you to quickly navigate between source files. Pressing Ctrl+Left-Click or F3 on any expression language (EL) declaration will navigate you to the corresponding java source.

CDI Tools offers you a list of navigation alternatives for the bean that will be injected.



Figure 6.1. Example of navigation alternatives

If multiple eligible beans are detected for the injection, you will see an option to Show All Assignable Beans.



Figure 6.2. Show All Assignable Beans

Selecting this option will display a dialog containing all eligible beans.

@Inject Object o



Figure 6.3. Multiple eligible beans

Corresponding classes and interfaces that are mentioned in beans.xml can be opened through pressing Ctrl+Left-Click on the type name.

Observers and Events can be navigated to through the same method as well; **Show CDI Observer Methods** and **Show CDI Events**.

CDI Observer Methods

MyGenericBean2.myObserver() - org.jboss.generic

MyGenericBean2.myObserver2() - org.jboss.generic





Figure 6.5. Events example

Producer and Disposer methods, if available, will appear in the context menu for an injection.

 $\mathbf{\nabla}$



Figure 6.6. Producer example



Figure 6.7. Disposer example

Appendix A. Revision History

Revision History Revision 1-0 Initial creation of book

Fri Jan 06 2012

lsaacRooskov<irooskov@redhat.com>