Liberica JDK Using in your IDE



Liberica JDK Revision 1.0 February 2025



Copyright © BellSoft Corporation 2018-2025.

BellSoft software contains open source software. Additional information about third party code is available at <u>https://bell-sw.com/third_party_licenses</u>. You can also get more information on how to get a copy of source code by contacting <u>info@bell-sw.com</u>.

THIS INFORMATION MAY CHANGE WITHOUT NOTICE. TO THE EXTENT PERMITTED BY APPLICABLE LAW, BELLSOFT PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL BELLSOFT BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF BELLSOFT IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in this document is governed by the applicable license agreement, which is not modified in any way by the terms of this notice.

Alpaquita, Liberica and BellSoft are trademarks or registered trademarks of BellSoft Corporation. The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis. Java and OpenJDK are trademarks or registered trademarks of Oracle and/or its affiliates. Other trademarks are the property of their respective owners and are used only for identification purposes.

Contents

1. Introduction				
Which Liberica to choose	5			
Liberica JDK and JFX	5			
2. IntelliJ IDEA	7			
Specifying Liberica JDK as default SDK for new projects	7			
Specifying Liberica JDK for a new project	9			
Specifying Liberica JDK for an existing project	10			
3. Visual Studio Code	12			
Prerequisites	12			
Specifying Liberica JDK as the SDK for a project				
4. Apache NetBeans	15			
Specifying Liberica JDK as default JDK	15			
Specifying Liberica JDK for a project				

5. Eclipse

Specifying Liberica JDK during Eclipse installation	20
Specifying Liberica JDK as default JRE for new projects	21
Selecting Liberica JDK when creating a project	21
Specifying Liberica JDK for an existing project	22

20

1. Introduction

To develop Java applications in your development environment or IDE, you need a Java SDK (JDK). A JDK is a software package that contains libraries, tools for developing and testing Java applications (development tools), and tools for running applications on the Java platform (Java Runtime Environment – JRE).

<u>Liberica JDK</u> is a secure and reliable tool for application development that is compatible with the broadest range of system configurations on the market. You can use Liberica JDK in your IDE to create Java applications.

In this document, you will learn how to use Liberica JDK in your IDE. For the general installation instructions, see Liberica JDK Installation Guide.

Which Liberica to choose

It depends on your needs, which Liberica JDK version to choose for your project.

If you develop from scratch, we recommend using the latest Liberica JDK LTS builds, which are guaranteed to have prompt bug fixes, security updates, and other improvements as needed. We keep supporting LTS Liberica JDK builds even after the corresponding versions of OpenJDK are no longer supported. For more information, see <u>Support Roadmap for Liberica JDK</u>.

If you run older software, stick to the version it was developed for. In case your app is based on the version that is no longer supported, download the safest build of this version and get ready to upgrade your app, because your data is no longer safe.

If you plan to use the latest language features, use the latest versions and builds of Liberica JDK. We release new builds concurrently with Oracle JDK. In addition, we put out security patches and customer fixes on a regular basis.

Liberica JDK and JFX

JFX is an open-source platform containing graphics and media tools for developing and deploying rich client applications. We deliver a special flavor of Liberica JDK (Full version) containing LibericaFX — our instance of OpenJFX. Liberica for Embedded also comes with LibericaFX enabling the developers to create GUIs for embedded systems.

If you plan to create desktop, web, mobile, or embedded systems applications with JFX, consider the following:

- If you develop your JavaFX applications using a build system, such as Maven, download the Standard version of Liberica JDK. Specify the modules and the versions you want in the pom.xml file, and the build system will download the required modules, including the native libraries for your platform.
- If you want to use the JFX SDK to develop your applications, download and install the Full version of Liberica JDK that contains LibericaFX.

2. IntelliJ IDEA

IntelliJ IDEA supports a wide range of Java versions for developing applications. See <u>Supported Java</u> <u>versions and features</u> for details.

Specifying Liberica JDK as default SDK for new projects

You can set up IntelliJ IDEA to use Liberica JDK for all new projects.

- 1. Start IntelliJ IDEA and click File > New Project Setup > Structure.
- 2. Select Project under Project Settings in the left pane.

Project Structure for New	Projects		×
$\leftarrow \rightarrow$			
Project Settings	SDK:	📜 liberica-17 BellSoft Liberica version 17.0.10 👻 🛛 Edit	
Project			
Libraries	Language level:	📑 liberica-17 BellSoft Liberica version 17.0.10	*
Platform Settings SDKs	Compiler output:	🔚 liberica-21 BellSoft Liberica version 21.0.2	-
Global Libraries		+ Add SDK >	g sources.
		Detected SDKs	
Problems		🖶 C:\Program Files\BellSoft\LibericaJDK-21 BellSoft Liberica version 21.0.1	
		📜 C:\Program Files\Java\jdk-18.0.1.1 Oracle OpenJDK version 18.0.1	
		📜 C:\Program Files\BellSoft\LibericaJDK-11 BellSoft Liberica version 11.0.20	

The right pane now contains project options including the selected SDK for all new projects.

- 3. Click the **SDK** list and select the required Liberica JDK version.
 - If Liberica JDK is installed on your computer, but not defined in the IDE, either select it from the detected SDKs, or click Add SDK > JDK, and specify the path to the JDK home directory, for example, C:\Program Files\BellSoft\LibericaJDK-21.

Select Home Directory for JDK	×
♠⊒ ka ka ka xa G @a	Hide path
C:\Program Files\BellSoft\LibericaJDK-21	-
Min C:	
Documents and Settings	
> PerfLogs	
🗡 🖿 Program Files	
> T-Zip	
> 🖿 Adobe	
Application Verifier	
🕆 🖿 BellSoft	
LibericaJDK-11	
LibericaJDK-17	
LibericaJDK-21	
ChromiumTemp5772_1977816449	
Common Files	
> Docker	
Drag and drop a file into the space above to quickly locate it	
? Ок	Cancel

- If Liberica JDK is not installed, you can download and install it right in the **New Project** window as follows:
 - a. Click the **JDK** list and select **Add SDK** > **Download JDK**.



b. In the **Download JDK** dialog box, specify a version, vendor, and change the installation path if necessary.

U Down	nload JDK	×
Version:	21	•
Vendor:	BellSoft Liberica JDK 21.0.2	•
Location:	C:\Users\js\.jdks\liberica-21.0.2-1	-
	Download	Cancel

c. Click Download.

Specifying Liberica JDK for a new project

If you have not specified Liberica JDK as an SDK for all new projects, you can select Liberica JDK in your new Java project while creating it in the IDE.

- 1. Start IntelliJ IDEA and click File > New > Project.
- 2. On the **New Project** screen, specify the name of your project, language (Java), and select Liberica JDK in the **JDK** list. The list contains the default JDK as well as detected JDKs installed on your computer.

Project	×
Q	
New Project	Name: MyJavaApp
Empty Project	Location: ~\ldeaProjects
Generators	Project will be created in: ~\ldeaProjects\untitled1
Maven Archetype	Create Git repository
JavaFX	Language: Java Kotlin Groovy HTML +
Kotlin Multiplatform	
Compose for Desktop	Build system: IntelliJ Maven Gradle
🕛 IDE Plugin	IDK:
🛎 Android	liberies 17 BallSaft Liberies version 17 D 6
	Add sample c
	Generate c + Add SDK
	Detected SDKs
	Advanced Setti C:\Program Files\Java\jdk-18.0.1.1 Oracle OpenJDK version 18.0.1
	C:\Program Files\BellSoft\LibericaJDK-11 BellSoft Liberica version 11.0.20
?	<u>C</u> reate Cancel

- If Liberica JDK is already defined in IntelliJ IDEA, select it from the SDK list.
- If Liberica JDK is installed on your computer, but not defined in the IDE, either select it from the detected SDKs, or click Add SDK > JDK, and specify the path to the JDK home directory, for example, C:\Program Files\BellSoft\LibericaJDK-21.
- If Liberica JDK is not installed, you can download and install it using the Add SDK > Download JDK option.

Specifying Liberica JDK for an existing project

Follow the next steps to use Liberica JDK in your existing Java project.

- 1. Start IntelliJ IDEA and open your Java project.
- 2. Select the project and click **File** > **Project Structure**.
- 3. Select Project under Project Settings in the left pane.

Project Structure	l.		×
$\leftarrow \rightarrow$			
Project Settings	Project		
Project	Default settings fo	r all modules. Configure these parameters for each module on the module page as needed.	
Modules	Name:	MyJavaApp	
Libraries			
Facets	SDK:	Iberica-17 BellSoft Liberica version 17.0.6 ▼ Edit	
Artifacts			
Platform Settings	Language level:	SDK default 🔹	
SDKs			
Global Libraries	Compiler output:	~\IdeaProjects\MyJavaApp\out	
Problems		Used for module subdirectories, Production and Test directories for the corresponding sources.	
?		OK Cancel	Apply

The right pane now contains your project options including the selected SDK.

4. Click the SDK list and select the required Liberica JDK version.

If Liberica JDK is not installed, you can download and install it using the **Add SDK** > **Download JDK** option.

3. Visual Studio Code

Prerequisites

We assume that you have set up Visual Studio Code (VSCode) to edit, run, and debug Java programs. For more information, see Java in Visual Studio Code.

To run Java within Visual Studio Code, you need to install a JDK. For the installation instructions, see <u>Liberica JDK Installation Guide</u>.

Note:

If you are working on Windows and macOS and you have installed the Coding Pack for Java, it already includes the Java Development Kit (JDK) and essential Java extensions.

Specifying Liberica JDK as the SDK for a project

VSCode uses the default JDK installed on your computer as an SDK for Java projects. Therefore, all you need to do to use the Liberica JDK as an SDK in VSCode, install it and check the default JDK in the command-line prompt as follows:

java -version openjdk version "21.0.1" 2023-10-17 LTS OpenJDK Runtime Environment (build 21.0.1+12-LTS) OpenJDK 64-Bit Server VM (build 21.0.1+12-LTS, mixed mode, sharing)

To see which JDKs are used for your projects, you can trigger the command Java: Configure Java Runtime in Command Palette (Ctrl+Shift+P). This command opens a view displaying the runtime information for your projects.



Click the Java Version list to see which JDKs are installed on your system and can be used for your project.

^{‡†‡} Settings	8 settings.json	☑ Configure Java Runtime ×	J App.java	⊟ …
Conf Manage J	igure Runt	time for Projects	a runtime, you can <u>d</u>	ownload one.
Project	Name	Туре	Java Version	
MyJava.	Арр	Unmanaged folder 🕠	21 - C:\Program 21 - C:\Program 18 - C:\Program 11 - C:\Program	Files V V Files\BellSoft\LibericaJDK-21 Files\Java\jdk-18.0.1.1 Files\BellSoft\LibericaJDK-11

If you have several JDKs installed on your computer and want to use Liberica JDK as an SDK in VSCode, perform the following:

- 1. Download and install the required version of Liberica JDK. See <u>Liberica JDK Installation Guide</u> for details.
- 2. Open the settings.json file in VSCode.
- 3. Add the java.configuration.runtimes setting specifying the required Liberica JDK version and correct paths.

```
{
    "java.project.sourcePaths": ["src"],
    "java.project.outputPath": "bin",
    "java.project.referencedLibraries": [
        "lib/**/*.jar"
],
    "java.configuration.runtimes": [
```



```
{
    "name": "JavaSE-21",
    "path": "C:\\Program Files\\BellSoft\\LibericaJDK-21",
    "sources" : "C:\\Program Files\\BellSoft\\LibericaJDK-
21\\libsrc.zip",
    "javadoc" : "https://docs.oracle.com/en/java/javase/21/docs/api/",
    "default": true
    }
]
}
Note:
You can set Liberica JDK to default by adding "default": true to the entry. The
default JDK will be used for your unmanaged folders.
```

For more information, see <u>Managing Java Projects in VS Code</u>.

4. Apache NetBeans

Apache NetBeans is a development environment that provides editors, wizards, and templates to help you create applications in Java, PHP and many other languages.

Specifying Liberica JDK as default JDK

When you install NetBeans, you are required to specify a JDK for the IDE during the installation process. This version of JDK is the default JDK used for all new Java projects.

Q Apache NetBeans IDE Installer	_		×
Apache NetBeans IDE 20 Installation Choose the installation folder and JDK™.		etBeans	IDE
Install the Apache NetBeans IDE to:			
C:\Program Files\NetBeans-20		Brows	e
JDK™ for the Apache NetBeans IDE:			
C:\Program Files\BellSoft\LibericaJDK-17		Brows	e
< <u>B</u> ack	<u>N</u> ext >	Can	cel

If you are installing NetBeans for the first time and want to use Liberica JDK for all of your Java projects, first <u>install Liberica JDK on your computer</u> and then choose Liberica JDK as the JDK for NetBeans IDE during the installation.



Specifying Liberica JDK for a project

If you want to use Liberica JDK in your project in NetBeans IDE that was installed with other default JDK, perform the following:

- 1. Download and install the required version of Liberica JDK. See <u>Liberica JDK Installation Guide</u> for details.
- 2. Start NetBeans.
- 3. Right-click your project in NetBeans and select **Properties** on the shortcut menu.
- 4. Select Libraries in the Categories pane.

Project Properties - AnagramGame		×
Categories:		
 Sources 	Java Platform: JDK 17 (Default)	Manage Platforms
Libraries	Libraries Folder:	Browse
 Compiling Packaging Deployment Documenting 	Compile Processor Run Compile Tests Run Tests Compile-time Libraries:	
♦ Run	Modulepath	+ Edit
	Classpath	+ Remove
Formatting		Move Up
 Hints 		Move Down
	Compile-time libraries are propagated to all library categories.	
	ОК	Cancel Help

5. Click Manage Platforms. The Java Platform Manager appears.

Use the Javadoc tab to register the API documentation for your JDK in the IDE. Click Add Platform to register other Java platform versions. Platforms: Platform Name: JDK 17 (Default) Platform Folder: C:\Program Files\BellSoft\LibericaJDK-17 Classes Sources Javadoc JDK Modules: java.dstatransfer java.dstatransfer java.dstatransfer java.dstatransfer java.dstatransfer java.dstatransfer java.dstatransfer java.dstatransfer java.dstatransfer java.nentt java.nentt java.nenttp java.nenttp java.nenttp java.nenttp java.nenttp java.nenttp java.nenttp	🟮 Java Platform Manager		×
Platforms: JDK 17 (Default) ■ JDK 17 (Default) Platform Folder: CLasses Sources JDK Modules: java.base java.desktop java.desktop java.nentment java.nengement java.nagement. java.nagement. java.nengement. java.nengement. java.nengement.	Use the Javadoc tab to register the API Click Add Platform to register other Jav	documentation for your JDK in the IDE. va platform versions.	
Java SE Platform Name: JDK 17 (Default) Platform Folder: C:\Program Files\BellSoft\LibericaJDK-17 Classes Sources Javadoc JDK Modules: java.abse java.desktop java.logging java.namagement.mii java.namagement.mii java.net.http java.net.http java.prefs Intervention 	Platforms:		
JDK 17 (Default) Platform Folder: C:\Program Files\BellSoft\LibericaJDK-17 Classes Sources Javadoc JDK Modules:	🗁 Java SE	Platform Name: JDK 17 (Default)	
Platform Folder: C:\Program Files\BellSoft\LibericaJDK-17 Classes Sources Javadoc JDK Modules: java.base java.compiler java.datatransfer java.desktop java.logging java.logging java.management java.management.rmi java.naming java.net.http java.prefs	JDK 17 (Default)		_
Classes Sources Javadoc JDK Modules: java.base java.compiler java.datatransfer java.desktop java.desktop java.logging java.logging java.management.rmi java.management.rmi java.naming java.net.http java.prefs		Platform Folder: C:\Program Files\BellSoft\LibericaJDK-17	
JDK Modules: java.base java.compiler java.datatransfer java.desktop java.instrument java.logging java.management.rmi java.management.rmi java.naming java.net.http java.prefs		Classes Sources Javadoc	
 java.base java.compiler java.datatransfer java.desktop java.instrument java.logging java.management java.management.rmi java.naming java.net.http java.prefs 		JDK Modules:	
 java.compiler java.datatransfer java.desktop java.instrument java.logging java.management java.management.rmi java.naming java.net.http java.prefs 		🎽 java.base	
 java.datatransfer java.desktop java.instrument java.logging java.management java.management.rmi java.naming java.net.http java.prefs 		🎬 java.compiler	
 java.desktop java.instrument java.logging java.management java.management.rmi java.naming java.net.http java.prefs 		🎥 java.datatransfer	
 java.instrument java.logging java.management java.management.rmi java.naming java.net.http java.prefs 		🎥 java.desktop	
 java.logging java.management java.management.rmi java.naming java.net.http java.prefs 		🎬 java.instrument	
 java.management java.management.rmi java.naming java.net.http java.prefs 		🎬 java.logging	
 java.management.rmi java.naming java.net.http java.prefs 		🎬 java.management	
 java.naming java.net.http java.prefs 		👺 java.management.rmi	
 java.net.http java.prefs 		java.naming	
2° java.prets		2 java.net.http	
		2 java.prefs	
Add Platform Remove	Add Platform Remove		
Close Help		Close He	lp

- 6. Click Add Platform.
- 7. Select Java Standard Edition on the Select platform type step and click Next.
- 8. On the **Choose Platform Folder** step, select a folder to look in and then select the Liberica JDK installation folder. Click **Next**.

🗘 Add Java Platform			×
Steps	Choose Platform	n Folder	
 Select platform type Choose Platform Folder Platform Name 	Recent Items Desktop Documents	Look In:	BellSoft V
	This PC	File Name: Files of Type:	C:\Program Files\BellSoft\LibericaJDK-21 Platform Folder
			< Back Next > Finish Cancel Help

9. The wizard detects the platform name, path to sources, and adds a link to the platform's Javadoc. If the required information is not detected, enter it manually. Click **Finish** to exit the wizard.

🗘 Add Java Platform		×
Steps	Platform Name	
Select platform type Choose Platform Folder Platform Name	Platform Name: JDK 21	
	Platform Sources:	Browse
	Platform Javadoc: https://docs.oracle.com/en/java/javase/21/docs/api/	Browse
	< Back Next > Finish Cancel	Help

The added Liberica JDK version should be listed under **Platforms** in the Java Platform Manager. Close the manager.

🔾 Java Platform Manager		
Use the Javadoc tab to register the AP Click Add Platform to register other Ja	l documentation for va platform version	your JDK in the IDE. s.
Platforms:		
🗁 Java SE	Platform Name:	JDK 21
🔜 JDK 17 (Default) 🔜 JDK 21	Platform Folder:	C:\Program Files\BellSoft\LibericaJDK-21

10. Now you can select the JDK for the project in the **Java Platform** list on the **Project Properties** window.

Q	Pro	ject Properties - AnagramGame				×
Cat	ego	ries:				
	٢	Sources	Java Platform:	JDK 21	\sim	Manage Platforms
	$^{\circ}$	Libraries	Charles Falder	JDK 17 (Default)	٦ř	Denues
\sim	٥	Build	Libraries Folder:	JDK 21	1	Browse
		 Compiling 			_	
		 Packaging 	Compile P	rocessor Run Compile Tests Run Tests		
		 Deployment 				
		 Documenting 	Compile-time	Libraries:		
	0	Run	Modulepath			+ Edit
l~	0	Application	G		ň	
		 Web Start 	Classpath			* Remove

5. Eclipse

The Eclipse IDE is famous for its Java Integrated Development Environment (IDE), but also includes a number of other tools including C/C++ IDE, JavaScript/TypeScript IDE, PHP IDE, and more.

Specifying Liberica JDK during Eclipse installation

When you install Eclipse, you are asked to specify a JVM to use with the IDE. The latest Eclipse versions require Java version 17 or later.

eclipseins	staller by Comph	. 0
Eclipse II The essent Editor, May	IDE for Java Developers details tial tools for any Java developer, including a Java IDE, a Git client, XML ven and Gradle integration.	
Java 17+ VM	C:\Program Files\BellSoft\LibericaJDK-21 🔹 🍃	
Installation Folder	C:\Program Files\eclipse\java-2023-12	
	 create start menu entry create desktop shortcut 	
	🕹 INSTALL	
< ВАСК		

If you are installing Eclipse for the first time and want to use Liberica JDK as the default JRE for all of your Java projects, first <u>install Liberica JDK on your computer</u> and then choose Liberica JDK as the JDK for Eclipse IDE during the installation.



Specifying Liberica JDK as default JRE for new projects

You can set up Eclipse IDE to use Liberica JDK for all new projects by selecting it in the preferences.

- 1. Start Eclipse IDE.
- 2. Go to Window > Preferences.
- 3. Select Java > Installed JREs in the navigation pane on the left.

/pe filter text	Installed JREs			$(\bigcirc \bullet \circ) \bullet \bullet$
General Ant	Add, remove or edit JRE definitions.	By default, the checked JRE is added to the buil	ld path of newly crea	ted Java project
Gradle	Detect available JVM installations	at startup		
Help	Installed IREs			
Install/Update	Name States	t	Tere	
/ Java	Name	Location	lype	<u>A</u> aa
> Appearance	LibericaJDK-17 (default)	C:\Program Files\BellSoft\LibericaJDK-17	Standard VM	Edit
> Build Path	LibericaJDK-21	C:\Program Files\BellSoft\LibericaJDK-21	Standard VM	-
Bytecode Outline				Duplicate
Code Coverage				
> Code Style				Remove
> Compiler				Search
> Debug				
> Editor				
> Installed JKEs				
JUnit				
Launching				
Properties Files Editor				

4. Select a Liberica JDK version under **Installed JREs** to make it the default JRE for new projects. Click **Apply and Close** for the changes to take effect.

If the installed Liberica JDK is not listed, click **Add** and follow the steps of the Wizard to create a new JRE definition in the IDE. Once it is added to the **Installed JREs** list, select it to make it the default JRE for new projects.

For more information, see Installed JREs Preferences in Eclipse Help.

Selecting Liberica JDK when creating a project

- 1. Download and install the required version of Liberica JDK. See <u>Liberica JDK Installation Guide</u> for details.
- 2. Start Eclipse IDE.

- 3. Select File > New > Java Project. You are prompted to specify a JRE to use with the project.
 - If Liberica JDK is selected as an execution environment JRE in Eclipse IDE, select **Use an** execution environment JRE.
 - If you specified other JDK as the execution environment JRE for Eclipse, select **Use a project-specific JRE** and select the Liberica JDK version you want to use. The new project will use a compiler compliance, which matches the version of the selected JRE.
 - If Liberica JDK is selected as the default JRE in Eclipse IDE, select Use default JRE. The New Java Project Wizard creates a new Java project, which uses the workspace default JRE. Click Configure JREs to add, remove, or edit JRE definitions.

New Java Project		- D X
Create a Java Project Enter a location for the project.		
Project name: MyJavaApp2		
Use default location		
JRE		
 Use an execution environment JRE: Use a project specific JRE: 	JavaSE-17 LibericaJDK-21	~ ~
O Use default JRE 'LibericaJDK-17' and v	LibericaJDK-21 LibericaJDK-17	
Project layout		
○ Use project folder as root for sources	and class files	
Create separate folders for sources ar	d class files	Configure default

For more information, see New Java Project Wizard in Eclipse Help.

Specifying Liberica JDK for an existing project

If you want to use Liberica JDK in your existing project in Eclipse IDE, perform the following:

- 1. Download and install the required version of Liberica JDK. See <u>Liberica JDK Installation Guide</u> for details.
- 2. Start Eclipse IDE.
- 3. Right-click your project in Eclipse and select **Properties** on the shortcut menu. Properties window for your project appears.



- 4. Select Java Build Path in the left pane and select the Libraries tab in the right pane.
- 5. Select the current JRE under **Modulepath** in the tree and click **Edit**.

Properties for MyJavaApp		– 🗆 X
type filter text	Java Build Path	⇔ - ⇔ * 8
 > Resource Builders Coverage Java Build Path > Java Code Style > Java Compiler > Java Compiler > Annotation Processing Building Errors/Warnings Javadoc Task Tags Javadoc Location > Java Editor Project Natures Project Natures Project References Rur/Debug Settings > Task Repository WikiText 	Source Projects Libraries On Order and Export O Module Dependencies JARs and class folders on the build path:	Add JARs Add External JARs Add Yariable Add Library Add Class Folder Add External Class Folder Edit <u>E</u> dit <u>R</u> emove <u>M</u> igrate JAR File
?	Apply	v and Close Cancel

6. In the Edit Library window, select the Liberica JDK version you want to use and click Finish.

🖨 Edit Library		-	_		×
JRE System Library Select JRE for the project build	l path.			J.	R
System library					
O Execution environment:	CDC-1.0/Foundation-1.0 (LibericaJDK-17)	~	Envir	onment	s
Alternate JRE:	LibericaJDK-17	~	Insta	lled JRE	s
O Workspace default JRE (Li	LibericaJDK-17				
?	Finish	1		Cance	4



7. Click Apply and Close in the properties window for your project.

For more information, see *Working with JREs* in Eclipse Help.



Liberica JDK Using in your IDE

