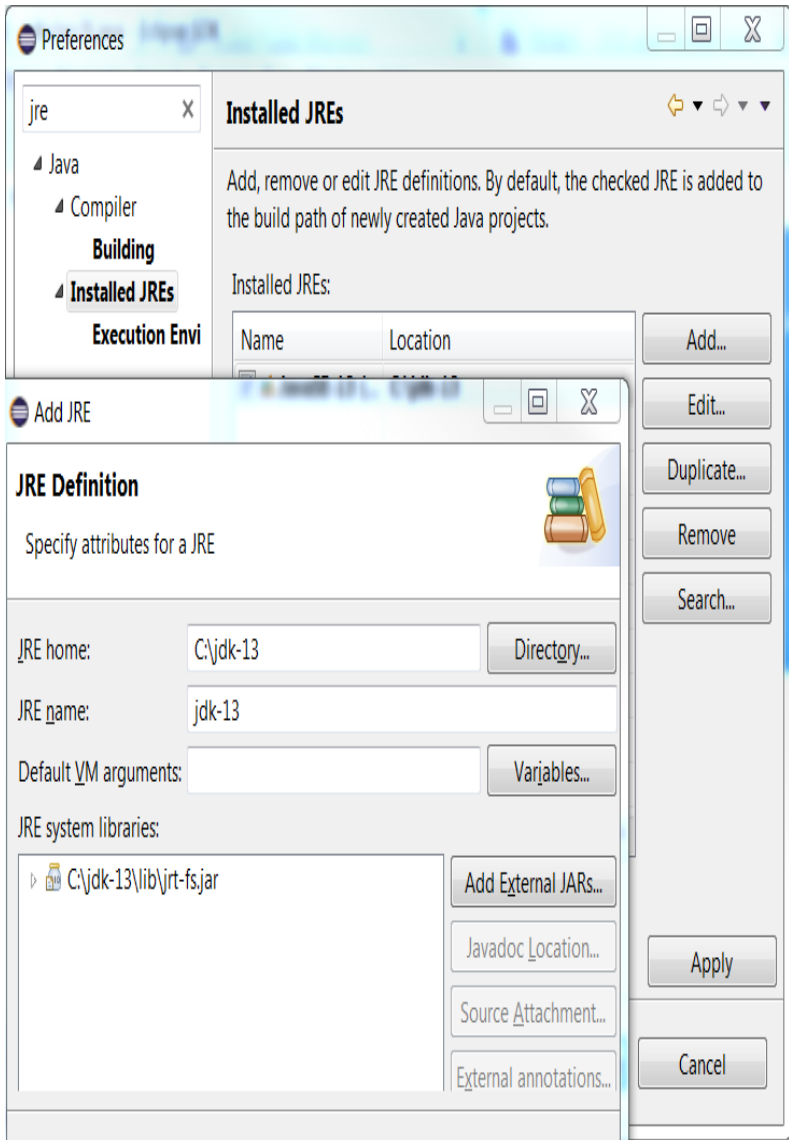


Java13/Examples

This is an informal page listing examples of features that are implemented by the Java 13 Support. You are welcome to try out these examples. If you find bugs, please file a bug after checking for a duplicate entry here (<http://eclip.se/fP>)

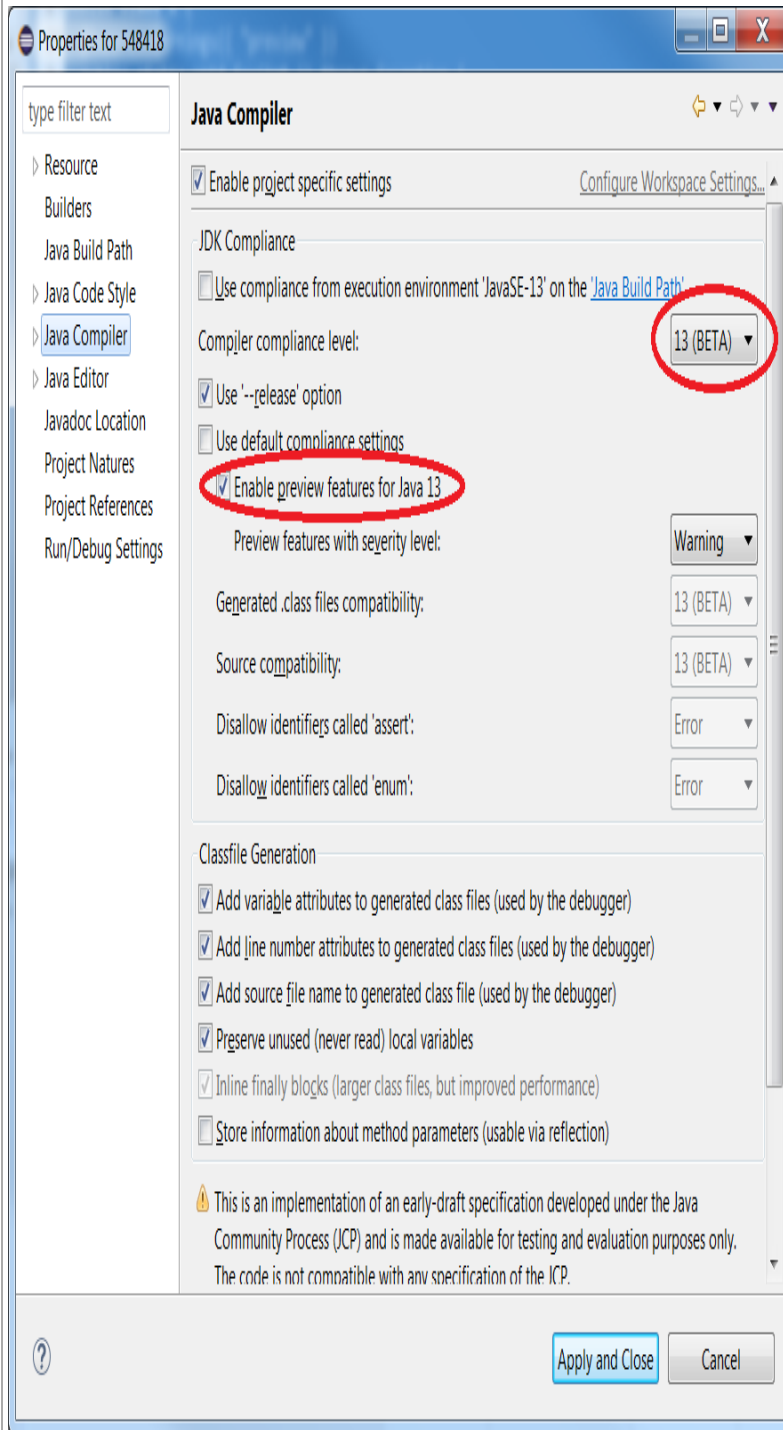
NOTE:

- Switch expression, Enhanced switch statement and Multi-constant case labels are **Preview features** in Java 13. They are not enabled by default and can be enabled using **--enable-preview**.
- TextBlock is also another preview feature in Java 13.
- In Eclipse, **--enable-preview** can be enabled from the Preferences. It is implicitly added while launching a java program if the feature has been enabled for the project/workspace.

	Feature / Steps	Expected Result
The Pre-requisite: Java 13 JRE Support		
Add Java 13 JRE	<p>Use Window -> Preferences-> Java -> Installed JREs -> Add...</p>  <p>(/File:FileAddJ13.png) [note: Eclipse -> Preferences in Mac / Window -> Preferences in Windows]</p>	Java 13 JRE recognized as a valid JRE
Project JRE	In Package Explorer Use project's context menu and add Java 13 JRE	JRE specific (eg Object) gets resolved in the project.
Package Explorer	Go to Package Explorer and expand the Java 13 JRE	Modules (eg java.base etc) are listed in the package explorer view
The First Step: Java 13 Compliance		

Context Menu of Project -> Properties -> Set project-specific, drop down to 13

Set Project Compliance in Package Explorer



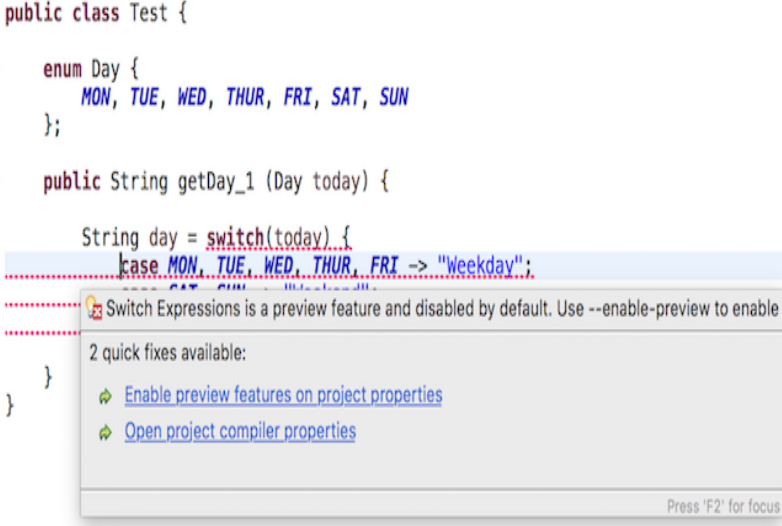
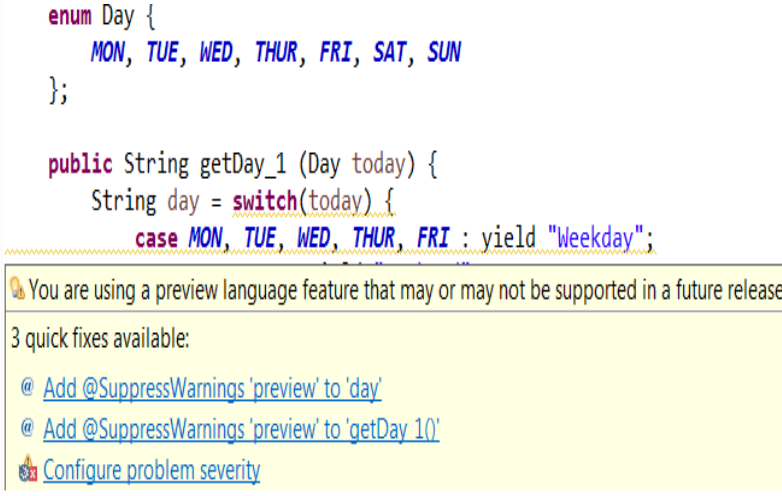
(/File:J13.compliance.png)

13 is shown in the drop down list.
A checkbox to enable preview features is available on the preference page.

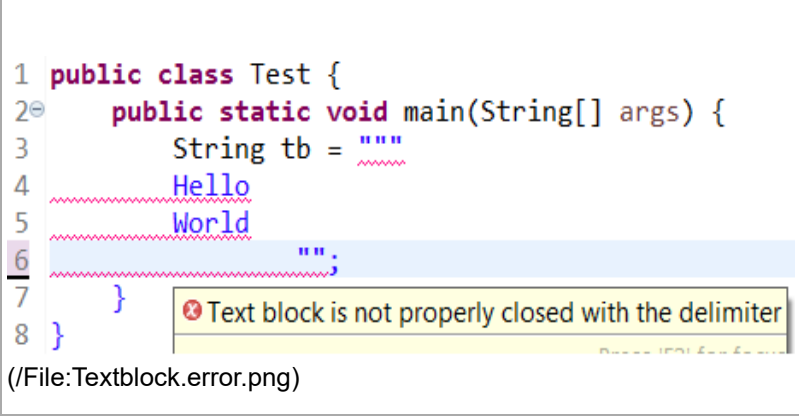
Preview Feature: Switch Expressions, Enhanced Switch Statement and Multi-Label Case Statements.

<p>Positive Compilation 1 (Switch Statement with multi-label case with colon)</p>	<p>Use the following code:</p> <pre> public class X { public void foo(int i) { switch (i) { case 0, 1, 2: System.out.println("Hello"); default : System.out.println("World"); } } public static void main(String[] argv) { new X().foo(2); } } </pre> <p>Code compiles (with a preview warning) and while running prints both "Hello" "World"</p>
<p>Positive Compilation 2 (Switch Statement with case with arrow)</p>	<p>Use the following code:</p> <pre> public class X { @SuppressWarnings("preview") public void foo(int i) { switch (i) { case 2 -> System.out.println("Hello"); default -> System.out.println("World"); } } public static void main(String[] argv) { new X().foo(2); } } </pre> <p>Code compiles (with a preview warning) and while running prints only "Hello" (because a break is implicit after every case with an arrow).</p>

<p>Positive Compilation (Switch Expression)</p>	<p>Use the following code:</p> <pre>public class Test { enum Day { MON, TUE, WED, THUR, FRI, SAT, SUN }; @SuppressWarnings("preview") public String getDay_1 (Day today) { String day = switch(today) { case MON, TUE, WED, THUR, FRI -> "Weekday"; case SAT, SUN -> "Weekend"; }; return day; } }</pre> <pre>public class Test { enum Day { MON, TUE, WED, THUR, FRI, SAT, SUN }; @SuppressWarnings("preview") public String getDay (Day today) { String day = switch(today) { case MON, TUE, WED, THUR, FRI -> "Weekday"; case SAT, SUN -> "Weekend"; }; return day; } }</pre> <p>(/File:Switch-exp.compile.png)</p>	<p>Code compiles</p>
---	---	----------------------

<p>Compiler error Case</p>	<pre> public class Test { enum Day { MON, TUE, WED, THUR, FRI, SAT, SUN }; public String getDay_1 (Day today) { String day = switch(today) { case MON, TUE, WED, THUR, FRI-> "Weekday"; case SAT, SUN->"Weekend"; }; return day; } } </pre>  <p>(/File:Enable-preview.error.png)</p>	<p>Compiler error is shown for preview feature. Quick fix is available to enable the preview feature in the preferences.</p>
<p>Compiler Warning Case</p>	<pre> public class Test { enum Day { MON, TUE, WED, THUR, FRI, SAT, SUN }; public String getDay_1 (Day today) { String day = switch(today) { case MON, TUE, WED, THUR, FRI : yield "Weekday"; case SAT, SUN : yield "Weekend"; }; return day; } } </pre>  <p>(/File:Switch.preview.warning.compile.yield.png)</p>	<p>Compiler warning is shown for preview feature</p>

<p>Compiler Error Case</p>	<pre> public class Test { enum Day { MON, TUE, WED, THUR, FRI, SAT, SUN }; @SuppressWarnings("preview") public String getDay_1 (Day today) { String day = switch(today) { case MON : yield "Weekday"; case SAT, SUN : yield "Weekend"; }; return day; } } </pre> <p>enum Day { MON, TUE, WED, THUR, FRI, SAT, SUN };</p> <p>@SuppressWarnings("preview") public String getDay_1 (Day today) { String day = switch(today) { case MON : yield case SAT, SUN : }; return day; }</p> <p>(/File:Switch.cover-all-cases.error.yield.png)</p>	<p>Compiler error is shown</p>
<p>Preview Feature: Text Blocks</p>		
<p>Text Block Example</p>	<p>Compile and run the following code:</p> <pre> @SuppressWarnings("preview") public class Test { public static void main(String[] args) { String tb = "" Hello World "" System.out.println(tb); } } </pre> <p>Test.java</p> <pre> 1 2 public class Test { 3 public static void main(String[] args) { 4 String tb = "" 5 Hello 6 World 7 ""; 8 System.out.println(tb); 9 } 10 } </pre> <p>(/File:Textblock.png)</p>	<p>Code compiles prints both "Hello" "World" as it is - notice that "World" is printed in the next line.</p>

<p>Text Block Compilation Error Example</p>	<p>Use the following code:</p> <div><pre>public class Test { public static void main(String[] args) { String tb = "" Hello World ""; } }</pre></div> 	<p>Compilation error - text block not closed properly</p>
--	---	---

This page was last modified 23:21, 15 September 2019 by Manoj Palat (/index.php?title=User:Manpalat.in.ibm.com&action=edit&redlink=1).

Copyright © Eclipse Foundation, Inc. All Rights Reserved.

