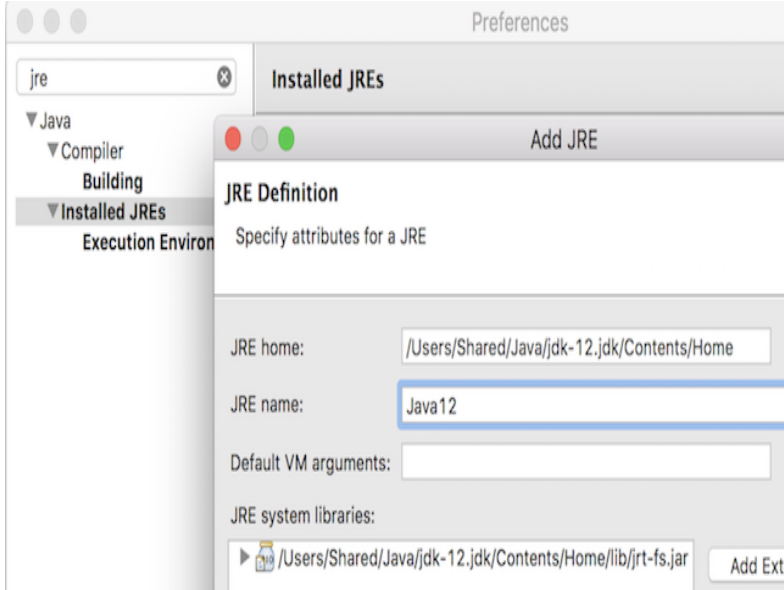
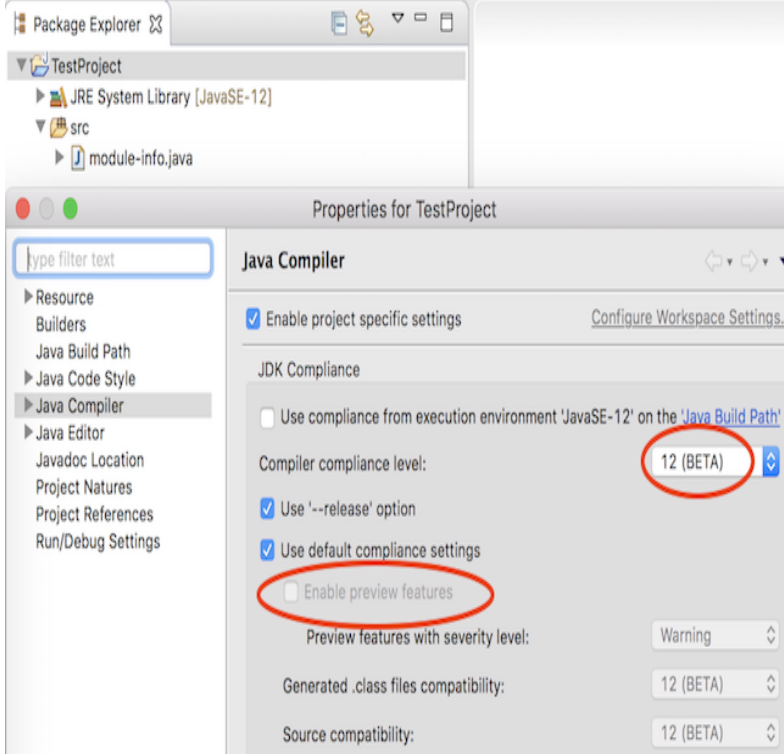


Java12/Examples

This is an informal page listing examples of features that are implemented by the Java 12 Support. You are welcome to try out these examples. If you find bugs, please file a bug after checking for a duplicate entry here (https://bugs.eclipse.org/bugs/buglist.cgi?cmdtype=dorem&list_id=18507270&namedcmd=J12.Open&remaction=run&sharer_id=152344)

NOTE:

- Switch expression, Enhanced switch statement and Multi-constant case labels are **Preview features** in Java 12. They are not enabled by default and can be enabled using **--enable-preview**.
- 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 12 JRE Support		
Add Java 12 JRE	<p>Use Window -> Preferences-> Java -> Installed JREs -> Add...</p>  <p>(/File:FileAddJ12.png) [note: Eclipse -> Preferences in Mac / Window -> Preferences in Windows]</p>	Java 12 JRE recognized as a valid JRE
Project JRE	In Package Explorer Use project's context menu and add Java 12 JRE	JRE specific (eg Object) gets resolved in the project.
Package Explorer	Go to Package Explorer and expand the Java 12 JRE	Modules (eg java.base etc) are listed in the package explorer view
The First Step: Java 12 Compliance		
Set Project Compliance in Package Explorer	<p>Context Menu of Project -> Properties -> Set project-specific, drop down to 12</p>  <p>(/File:J12.compliance.png)</p>	<p>12 is shown in the drop down list. A checkbox to enable preview features is available on the preference page.</p>

Basic Necessity : Compilation and Error Reporting

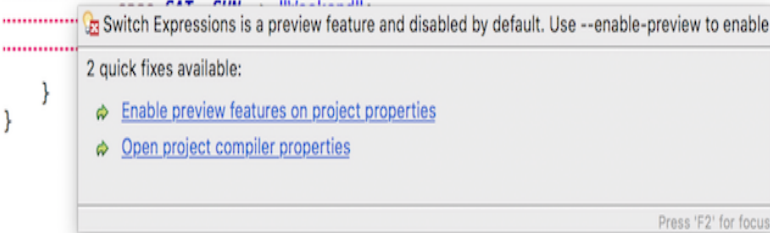
<p>Positive Compilation 1 (Switch Statement with multi-label case with colon)</p>	<p>Use the following code:</p> <pre>public class X { public void foo(int i) { switch (i) { case 0, 1, 2: System.out.println("Hello"); default : System.out.println("World"); } } public static void main(String[] argv) { new X().foo(2); } }</pre> <p>Code Snippet:</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>(/File:Switch-statement-multi.png)</p>	<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 Snippet:</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>(/File:Switch-statement-arrow.png)</p>	<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>
--	--	----------------------

**Compiler error
Case**



```
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: break "Weekday";  
            case SAT, SUN: break "Weekend";  
        };  
        return day;  
    }  
}
```

```
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;  
    }  
}
```



(/File:Enable-preview.error.png)


Compiler error is shown for preview feature. Quick fix is available to enable the preview feature in the preferences.

<p>Compiler Warning Case</p>	<div data-bbox="335 69 1129 436"><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: break "Weekday"; case SAT, SUN: break "Weekend"; }; return day; } }</pre></div> <div data-bbox="335 436 1129 1176"><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: break "Weekday"; case SAT, SUN: break "Weekend"; }; return day; } }</pre><div data-bbox="335 817 1129 1131"><p> You are using a preview language feature that may or may not be supported in a future release</p><p>3 quick fixes available:</p><ul style="list-style-type: none">@ Add @SuppressWarnings 'preview' to 'day'@ Add @SuppressWarnings 'preview' to 'getDay_1()' Configure problem severity<p>Press 'F2' for focus</p></div><p>(/File:Switch.preview.warning.compile.png)</p></div>	<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 : break "Weekday"; case SAT, SUN : break "Weekend"; }; return day; } } </pre> <pre> 3 public class Test { 4 5 enum Day { 6 MON, TUE, WED, THUR, FRI, SAT, SUN 7 }; 8 9 @SuppressWarnings("preview") 10 public String getDay_1 (Day today) { 11 A Switch expression should cover all possible values 12 case MON : break "Weekday"; 13 case SAT, SUN : break "Weekend"; 14 }; 15 return day; 16 } 17 } </pre> <p>(/File:Switch.cover-all-cases.error.png)</p>	<p>Compiler error is shown</p>
----------------------------	---	--------------------------------

This page was last modified 23:23, 19 March 2019 by Manoj Palat (/index.php?title=User:Manpalat.in.ibm.com&action=edit&redlink=1). Based on work by Sarika Sinha (/index.php?title=User:Sarika.sinha.in.ibm.com&action=edit&redlink=1) and Lakshmi Shanmugam (/index.php?title=User:Lshanmug.in.ibm.com&action=edit&redlink=1).

Copyright © Eclipse Foundation, Inc. All Rights Reserved.


<https://www.eclipse.org/clipse/>