

Working with Scala Implicit Conversions



Redirection Notice

This page will redirect to <https://www.jetbrains.com/help/idea/working-with-scala-implicit-conversions.html>.

Scala implicit conversions is a powerful tool for Scala coding. It lets you decrease redundancy in your code and lets you make the code easier to read.

Scala implicit conversions feature enables you to do the following:

- Highlight methods in which implicit conversions were used.
- Select from the available list of all applicable implicit conversions.
- Apply conversion either explicitly or in shorter version and still get the same result.
- Invoke implicit parameters.

Implicit Conversion highlighting

If Scala type-aware highlighting  is enabled then IntelliJ IDEA highlights methods that are added via implicit conversions with a grey underline:

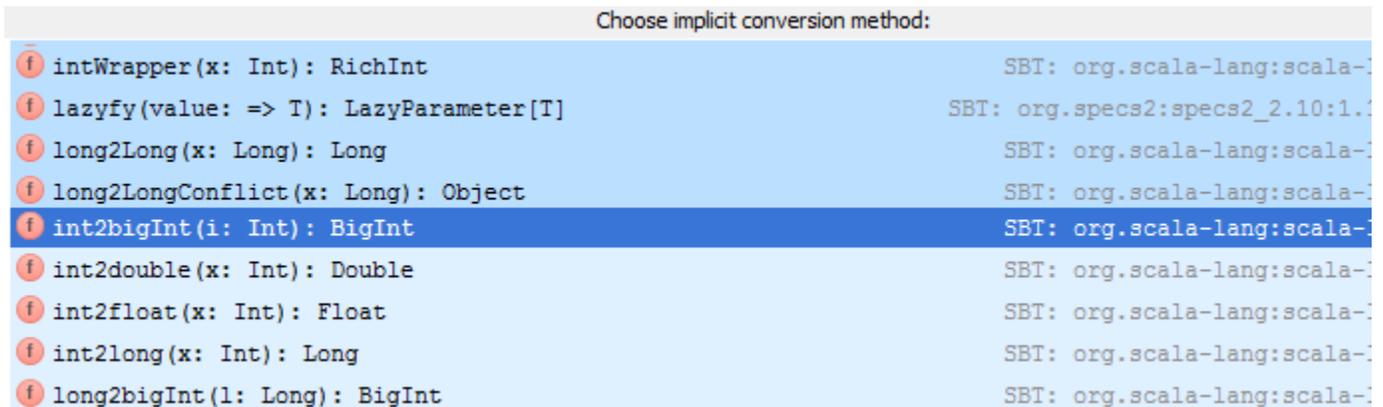
```
val x: BigInt = 1
```

Invoking implicit conversion methods

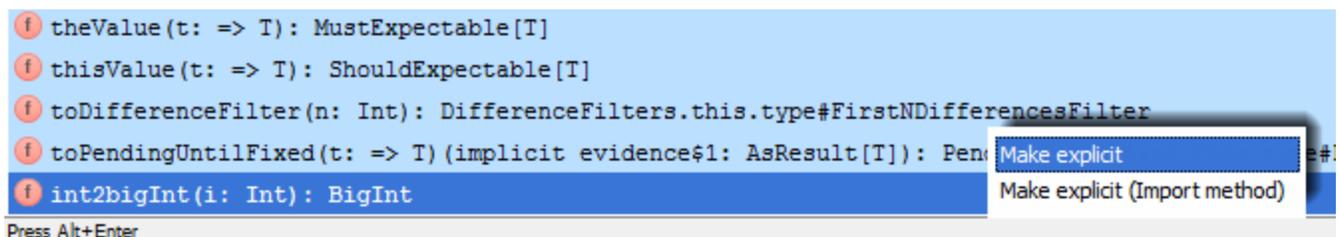
Select an expression and press Ctrl+Shift+Q (Ctrl+Q for Mac) to invoke the list of applicable implicit conversions. The list shows the regular scope displayed on the top and the expanded scope that is displayed on the bottom of the list.



Note: IntelliJ IDEA highlights an implicit conversion that was used for the selected expression. If IntelliJ IDEA cannot find the implicit conversion or if it finds more than one match then the list Introduce Variable opens.



You can make the implicit conversion method explicit. Press Alt+Enter and select Make explicit or Make explicit (Import method):



If you select Make explicit then IntelliJ IDEA returns a method call with the class name. It might be helpful if you need to make

sure that the compiler imports the particular implicit conversion method that you originally wanted:

```
val x: BigInt = math.BiGInt.int2bigInt(1)
```

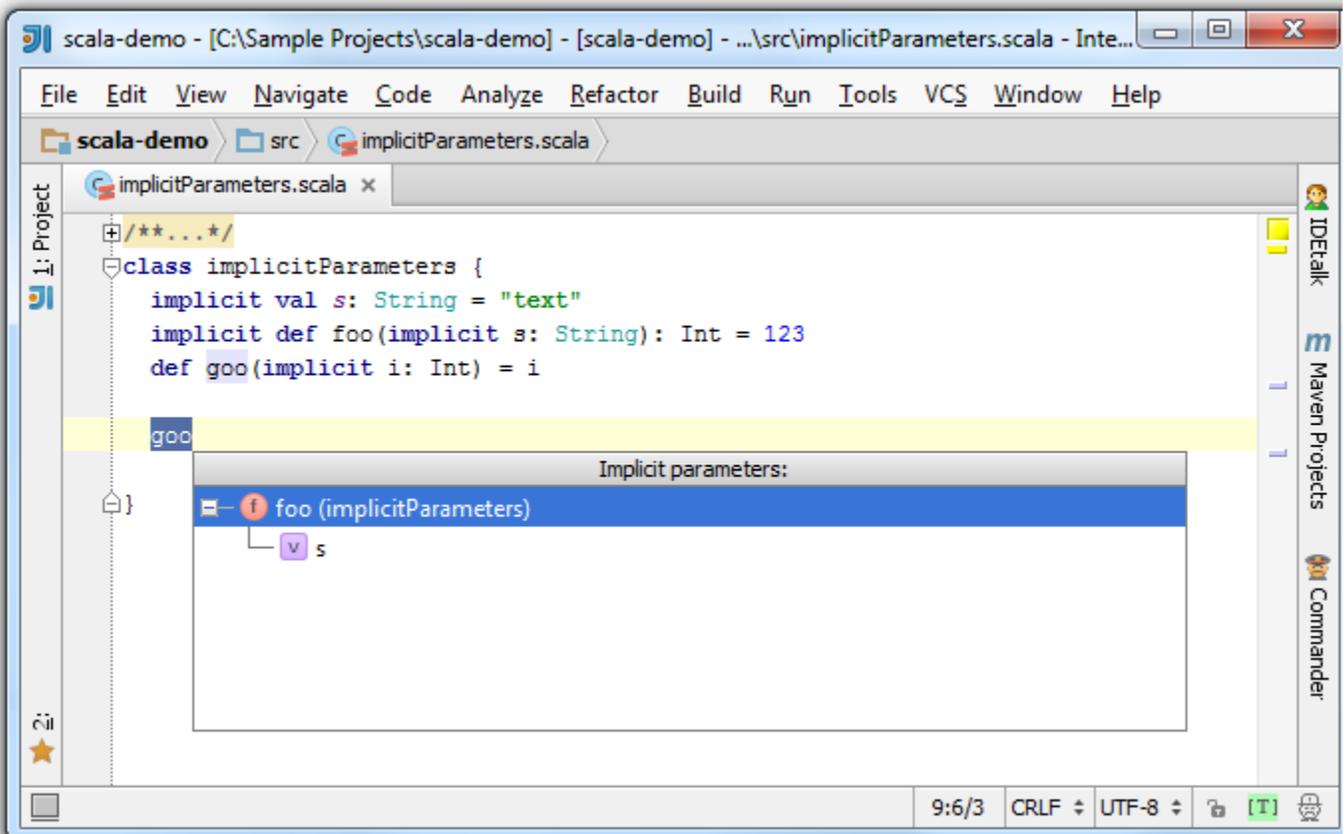
If you select Make explicit (Import method) then the method is imported statically and IntelliJ IDEA returns just its call without the class name. Also, the next time you open the list of useful implicit conversions you will see this method in the regular scope:

```
val x: BigInt = int2bigInt(1)
```

Invoking implicit parameters

Place a cursor to the method where implicit conversion was used and press Ctrl+Shift+P (Meta+Shift+P for Mac) to invoke implicit parameters. It might be helpful for code analyzing when you want to find out what implicit parameters were passed to the particular call. IntelliJ IDEA also lets you view the recursive implicit parameters.

 Note: IntelliJ IDEA highlights the method call where implicit parameters were used.



If IntelliJ IDEA cannot find method calls where implicit parameters were passed, it displays a pop-up message:

No implicit parameters



See also the tutorial [Getting started with Scala in IntelliJ IDEA](#).