

# Creating Method Breakpoints

---

In this section:

- [Basics](#)
- [Creating method breakpoints in the editor](#)
- [Creating method breakpoints using the Breakpoints dialog](#)
- [Deleting method breakpoints](#)

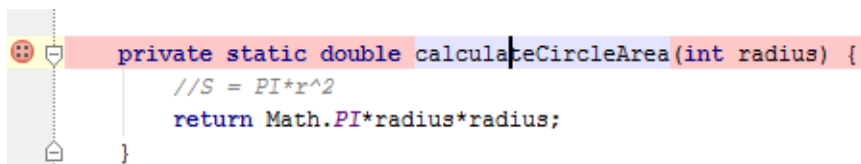
## Basics

**Method breakpoints** 🚫 let you follow the program flow at the method level.

### Creating method breakpoints in the editor

#### To create a method breakpoint using the editor

1. Place the caret inside a method, where you want to toggle a method breakpoint.
2. On the main menu, choose **Run | Toggle Method Breakpoint**. Method breakpoint appears at the method declaration.



A balloon appears, informing about the possible slow down of the debugging process:



Alternatively, just click the left gutter at the method declaration.

When one sets a breakpoint on a [folded method](#), a line breakpoint is set on the first executable line after method declaration.

### Creating method breakpoints using the breakpoints dialog

#### To create a method breakpoint using the Breakpoints dialog

1. On the main menu, choose **Run | View Breakpoints**, or press **Ctrl+Shift+F8**.
2. In the **Breakpoints** dialog box that opens, click **+**.
3. Select **Method Breakpoint** from the drop-down list.
4. In the **Add Method Breakpoint** dialog box, specify the class name pattern, including the package name, and the name of the desired method.

So doing, when the debugging session starts, the application will pause in all classes with the names matching the specified pattern, at the specified method.

### Deleting method breakpoints

#### To remove a method breakpoint, do one of the following

- Click the method breakpoint icon 🚫 in the left gutter.
- On the main menu, choose **Run | Toggle Method Breakpoint**.

### See Also

### Concepts:

- [Run/Debug Configuration](#)
- [Breakpoints](#)

### Reference:

- [Breakpoints](#)

### Web Resources:

- [Developer Community](#) 