



CONTENTS INCLUDE:

- JDK 7 Features
- Keyboard Shortcuts & Code Templates
- 10 Handy Editor Shortcuts
- Resources and more!



NetBeans IDE 7: Programming in Java 7

By Geertjan Wielenga

INTRODUCTION

NetBeans IDE 7 is an Oracle sponsored free and open-source Java integrated development environment. Developers from the Java Development Kit (JDK) team have worked closely with developers from the NetBeans team to create a well-aligned JDK 7 development experience for Java developers in NetBeans IDE.

NetBeans IDE 7 introduces support for new JDK 7 language features, such as the diamond syntax for constructors, strings in switch, and multicatch. When you use these constructs in your code, NetBeans IDE recognizes them, offers correct classes in code completion, correctly highlights errors, and lets you automatically migrate old syntax.

You'll find this reference card helpful if you want to get as much out of JDK 7 and NetBeans IDE 7 as their creators intended. This card focuses on all the new JDK 7 language features and on how NetBeans IDE 7 has been tuned to support them. In addition, you will find lists of NetBeans IDE keyboard shortcuts in carefully thought-out categories, together with expositions of optimal handling of Java code in the editor, covering viewing, navigation, source handling, and refactoring.

Go to <http://www.netbeans.org> to get NetBeans IDE 7.

JDK 7 FEATURES

This section introduces all the JDK 7 language enhancements in Project Coin and how you can start using them in NetBeans IDE 7.

- **Strings in Switch** – In previous versions of Java, the argument of “switch” had to be only of the primitive data types: byte, short, char, int, or enum. Now, with JDK 7, the String data type is added. NetBeans IDE 7 recognizes when if-then-else statements are used instead of switch statements and offers to convert them to the switch statement.

```

16 public static void main(String[] args) {
17     String color = "red";
18     colorRGB;
19     if (color.equals("black")) {
20         colorRGB="008000";
21     }
22     colorRGB="008000";
23 }
24

```

Annotations: "Convert to switch" over the if statement, "Convert to switch over Strings" over the if statement.

- **Diamond Syntax Support** – The new diamond syntax for constructors allows for more compact and readable code by saving repeated type arguments. Included is new support for LinkedList<>().

```

12 public static void main(String[] args) throws IOException {
13     HashMap<String, Integer> map = new HashMap<String, Integer>();
14     HashMap<> map2 = new HashMap<>();
15     if (a.equals("foo")) {
16         System.err.println("FOO");
17     }
18 }
19

```

Annotations: "Use diamond inference" over the second constructor call.

- **Integer literals and underscores** – Binary integer literals are now available in addition to octal, decimal, and hexadecimal literals. Underscores make code more readable.

```

int phoneNumber = 123_456_7890;
long creditCardNumber = 1234_5678_9012_3456L;
int socialSecurityNumber = 999_99_9999;
int hexEquivalentOfIntegerMAX = 0x7f_ff_ff_ff; // Integer
int hexBytes = 0xFF_EC_DE_5E; // Integer.toHexString(-125)
int hexWords = 0xFFEC_DE5E; // Integer.toHexString(-12537)
long maxLong = 0x7fff_ffff_ffff_ffffL; // Long.toHexString(Long.MAX_VALUE)
int weirdBitFields = 0b000_10_101; // Integer.toBinaryString(10)

```

- **Try-with-resources** – Correctly and robustly manage resources that otherwise need to be explicitly closed. This form of the try statement obviates the need for manual resource closing.

```

public static void main(String[] args) throws IOException {
    try (FileInputStream in = new FileInputStream("foo.txt")) {
        int k;
        while ((k = in.read()) != -1) {
            System.out.write(k);
        }
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}

```

- **Multicatch Variable Declaration Support** – Code completion is supported for the multicatch variable declaration. Hints to join multiple catches into one multicatch are provided if the handling code is the same.

```

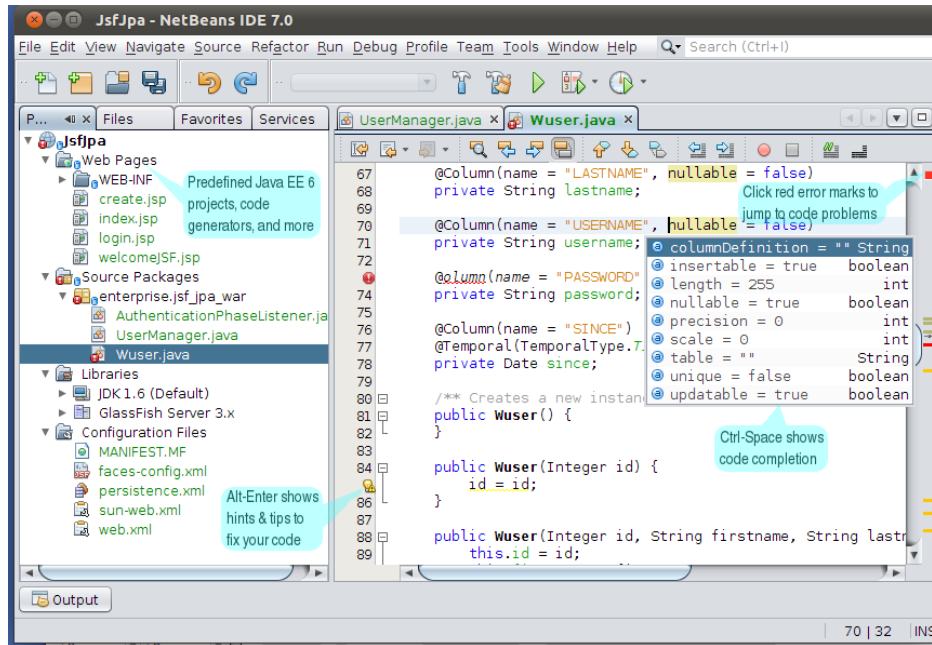
11 try {
12     throw new FileNotFoundException("adasdf");
13 } catch (FileNotFoundException | IOException fnfo) {
14 }
15
16

```

NetBeans IDE 7

NetBeans IDE 7 is an integrated development environment available for Windows, Mac, Linux, and Solaris that enables developers to rapidly create Java applications for web, enterprise, desktop, and mobile applications.

JAVA EDITOR OVERVIEW



NetBeans IDE 7 is a powerful and intuitive Java development environment.

To the left, you see the parts of the IDE that are most commonly used, in particular the rich code editor.

Combined with the popular Matisse GUI Builder, Profiler, Java debugger, and tools for generating Java EE 6 applications, the NetBeans IDE provides a complete & productive environment for developing all types of Java applications.

KEYBOARD SHORTCUTS & CODE TEMPLATES

Finding, Searching, and Replacing

Ctrl-F3	Search word at insert point
F3/Shift-F3	Find next/previous in file
Ctrl-F/H	Find/Replace in file
Alt-F7	Find usages
Ctrl-Shift-F/H	Find/replace in projects
Alt-Shift-U	Find usages results
Alt-Shift-H	Turn off search result highlights
Ctrl-R	Inplace Rename
Ctrl-U, then U	Convert selection to uppercase
Ctrl-U, then L	Convert selection to lowercase
Ctrl-U, then S	Toggle case of selection
Alt-Shift-V	Paste formatted

Opening and Toggling Between Views

Ctrl-Tab (Ctrl-)	Toggle between open documents
Shift-Escape	Maximize window (toggle)
Ctrl-F4/Ctrl-W	Close currently selected window
Ctrl-Shift-F4/Ctrl-Shift-W	Close all windows
Shift-F10	Open contextual menu
Alt-Shift-D	Undock window
Ctrl-4	Open Output window
Ctrl-8	Open Kenai Dashboard

Navigating through Source Code

Ctrl-O/Alt-Shift-O	Go to type/file
Ctrl-Shift-T	Go to related JUnit test, if any
Alt-O	Go to source

Ctrl-B	Go to declaration
Ctrl-G	Go to line
Ctrl-Shift-M	Toggle add/remove bookmark
Ctrl-Shift-Period/Comma	Next/previous bookmark
Ctrl-Period/Comma	Next/previous usage/compile error
Ctrl-Shift-1/2/3	Select in Projects/Files/Favorites
Ctrl-]	Move caret to matching bracket
Ctrl-K/Ctrl-Shift K	Next/previous word match
Alt-Left/Alt-Right/Ctrl-Q	Go backward/forward/to last edit

Compiling, Testing, and Running

F9	Compile package/ file
F11	Build main project
Shift-F11	Clean & build main project
Ctrl-Q	Set request parameters
Ctrl-Shift-U	Create JUnit test
Ctrl-F6/Alt-F6	Run JUnit test on file/project
F6/Shift-F6	Run main project/file

Debugging

Ctrl-F5	Start debugging main project
Ctrl-Shift-F5	Start debugging current file
Ctrl-Shift-F6	Start debugging test for file (JUnit)
Shift-F5/F5	Stop/Continue debugging session
F4	Run to cursor location in file
F7/F8	Step into/over
Ctrl-F7	Step out
Ctrl-Alt-Up	Go to called method
Ctrl-Alt-Down	Go to calling method
Ctrl-F9	Evaluate expression

Ctrl-F8	Toggle breakpoint
Ctrl-Shift-F8	New breakpoint
Ctrl-Shift-F7	New watch

Refactoring

This table provides short descriptions of the refactoring operations that are available in the IDE, mostly from under the Refactoring menu and within the Java editor itself when you right-click within it.

Refactoring Operation	Description
Rename	Enables you to change the name of a class, variable, or method to something more meaningful. In addition, it updates all source code in your project to reference the element by its new name.
Introduce Variable, Constant, Field, or Method	Enables you to generate a statement based on the selected code and replace that block of code with a call to the statement.
Change Method Parameters	Enables you to add parameters to a method and change the access modifier.
Encapsulate Fields	Generates a getter method and a setter method for a field and optionally updates all referencing code to access the field using the getter and setter methods.
Pull Up	Moves methods and fields to the superclass.
Push Down	Moves inner classes, methods, and fields to all subclasses of their current class.
Move Class	Moves a class to another package or into another class. In addition, all source code in your project is updated to reference the class in its new location.
Copy Class	Copies a class to the same or a different package.
Move Inner to Outer Level	Moves an inner class one level up in hierarchy.
Convert Anonymous Class to Inner	Converts an anonymous class to an inner class that contains a name and constructor. The anonymous inner class is replaced with a call to the new inner class.
Extract Interface	Creates a new interface from the selected public non-static methods in a class or interface.
Extract Superclass	Creates a new abstract class, changes the current class to extend the new class, and moves the selected methods and fields to the new class.
Use Supertype Where Possible	Changes code that references the selected class (or other type) to instead use a supertype of that type.
Safely Delete	Checks for references to a code element and then automatically deletes that element if no other code references it.

When typing in the Source Editor, you can generate the text in the right column of the following list by typing the abbreviation that is listed in the left column and then pressing Tab.

Java Editor Code Templates

En	Enumeration
Ex	Exception
Ob	Object
Psf	public static final
Psfb	public static final boolean
Psfi	public static final int
PsfS	public static final String
St	String
ab	abstract
bo	boolean
br	break
ca	catch (
cl	class

cn	continue
df	default:
dowhile	do { } while (condition);
eq	equals
ex	extends
fa	false
fi	final
fl	float
forc	for (Iterator it = collection.iterator(); it.hasNext();) { Object elem = (Object) it.next(); }
fore	for (Object elem : iterable) { }
fori	for (int i = 0; i < arr.length; i++) { }
fy	finally
ie	interface
ifelse	if (condition){}else { }
im	implements
iof	instanceof
ir	import
le	length
newo	Object name = new Object(args);
pe	protected
pr	private
psf	private static final
psfb	private static final boolean
psfi	private static final int
psfs	private static final String
pst	printStackTrace();
psvm	public static void main(String[] args){ }
pu	public
re	return
serr	System.err.println ("");
sout	System.out.println ("");
st	static
sw	switch (
sy	synchronized
tds	Thread.dumpStack();
th	throws
trycatch	try { catch (Exception e) {
tw	throw
twN	throw new
wh	while (
whileit	while (it.hasNext() { Object elem = (Object) it.next(); }

JSP Code Templates

ag	application.getAttribute("")
ap	application.putAttribute("",)
ar	application.removeAttribute("")
cfgi	config.getInitParameter("")

jspf	<jsp:forward page=" "/>
jspg	<jsp:getProperty name=" " property=""/>
jspj	<jsp:include page=" "/>
jspk	<jsp:plugin type=" " code="" codebase=""></jsp:plugin>
jspm	<jsp:setProperty name=" " property=""/>
jspn	<jsp:useBean id=" " type=""/>
oupp	out.print(" ")
oupl	out.println(" ")
pcgp	pageContext.getAttribute(" ")
pcgn	pageContext.getAttributeNamesInScope(" ")
pcgs	pageContext.getAttributesScope(" ")
pcr	pageContext.removeAttribute(" ")
pcs	pageContext.setAttribute(" ",)
pg	<%@page %>
pga	<%@page autoFlush="false"%>
pgb	<%@page buffer=" kb"%>
pgc	<%@page contentType=" "%>
pgerr	<%@page errorPage=" "%>
pgex	<%@page extends=" "%>
pgie	<%@page isErrorPage="true"%>
pgim	<%@page import=" "%>
pgin	<%@page info=" "%>
pgit	<%@page isThreadSafe="false"%>
pgl	<%@page language="java"%>
pgs	<%@page session="false"%>
rg	request.getParameter(" ")
sg	session.getAttribute(" ")
sp	session.setAttribute(" ",)
sr	session.removeAttribute(" ")
tglib	<%@taglib uri=" "%>

Changing Default Settings

Each code template can be customized and new ones can be easily created.

1. Choose **Tools > Options** from the main menu.
2. For **code templates**, select Editor and click the Code Templates tab. Here you can also change the **expansion** key from Tab to something else.
3. For **keyboard shortcuts**, select Keymap and choose a profile from the Profile drop-down list.

Mac OS Keyboard Shortcuts

In most cases, working with the IDE on the Mac is no different from working on other operating systems. Two significant differences do exist, however. First, the Options window on the Mac is found under **NetBeans > Preferences**. Second, the About box is under **NetBeans > About**.

Scrolling and Selecting

Keys	Action
Cmd-[Moves the insertion point to the highlighted matching bracket. Note that this shortcut only works when the insertion point is located immediately after the opening bracket.
Cmd-Shift-[Selects the block between a pair of brackets. Note that this shortcut only works when the insertion point is located immediately after either the opening or closing bracket.

Ctrl-G	Jumps to any specified line.
Cmd-A	Selects all text in the file.

Code Folding

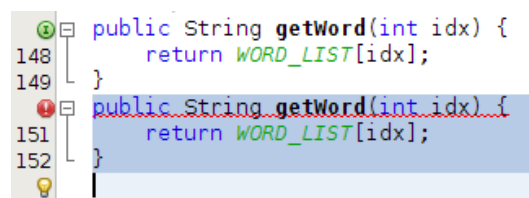
Keys	Action
Cmd-Minus (-)	Collapses the block of code in which the insertion point is currently located.
Cmd-Plus (+)	Expands the block of code which is adjacent to the insertion point.
Cmd-Shift-Minus (-)	Collapses all blocks of code in the current file.
Cmd-Shift-Plus (+)	Expands all blocks of code in the current file.

Cutting, Copying, Pasting, and Deleting Text

Keys	Action
Cmd-Z	Undo. Reverses a series of editor actions one at a time (excluding Save).
Cmd-Y	Redo. Reverses a series of Undo commands one at a time.
Cmd-X	Cut. Deletes the current selection and places it on the clipboard.
Cmd-C	Copy. Copies the current selection to the clipboard.
Cmd-V	Paste. Pastes the contents of the clipboard at the insert point.
Backspace Delete	Deletes the current selection.
Cmd-E	Deletes the current line.
Cmd-K	Copies the word preceding the insertion point and then pastes it after the insertion point (the insertion point must be in the whitespace preceding or following a word). Press K multiple times to cycle through preceding words in succession.
Cmd-Shift-K	Copies the word following the insertion point and pastes it at the insertion point (the insertion point must be located in the whitespace preceding or following a word.) Press L multiple times to cycle through consecutive following words.

10 HANDY EDITOR SHORTCUTS

1. **Move/copy up/down** – Press Ctrl-Shift-Up and the current selection is copied to the lines right above the current selection. Press Alt instead of Ctrl and it is moved instead of copied. Press Down instead of Up and the lines of code will be copied below the current selection.



```

148 public String getWord(int idx) {
149     return WORD_LIST[idx];
150 }
151 public String getWord(int idx) {
152     return WORD_LIST[idx];
153 }
    
```

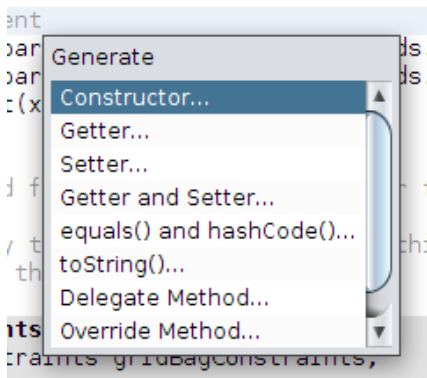
2. **Capture inner/outer syntactic element** – Each time you press Alt-Shift-Period, the selection expands to a successively wider syntactic element.

For example, one statement was selected below, the key combination Alt-Shift-Period was pressed, and then the complete content of the surrounding block statement was shown to be selected.

The selection expands from the current statement to surrounding block statements to the surrounding method and, from there, to the surrounding class and further. To successively narrow the selection, press Alt-Shift-Comma instead of Alt-Shift-Period.

```
/** Creates new form About */
public About(JFrame parent) {
    super(parent,true);
    initComponents();
    pack();
    Rectangle parentBounds = parent.getBounds();
    Dimension size = getSize();
    // Center in the parent
    int x = Math.max(0, parentBounds.x + (parentBo
    int y = Math.max(0, parentBounds.y + (parentBo
    setLocation(new Point(x, y));
}
```

3. **Generate code skeletons** – Whenever you want to generate commonly needed pieces of code, such as constructors, getters, and setters, simply click Alt-Insert and a small popup appears with a list of items from which you can select.



4. **Inplace rename** – If you want to change a variable, method, or other item of which more than one are used in your code, press Ctrl-R and you will see that all instances of the item turn blue at the same time. Now, when you change the selected item, all the other instances change at the same time until you press Escape, at which point the inplace rename mode comes to an end.

```
/** Creates new form About */
public About(JFrame parent) {
    super(parent,true);
    initComponents();
    pack();
    Rectangle parentBounds = parent.getBounds();
    Dimension size = getSize();
}
```

5. **Add/Remove comment lines** – Select one or more lines, press Ctrl-/ and then the selected lines are commented out.

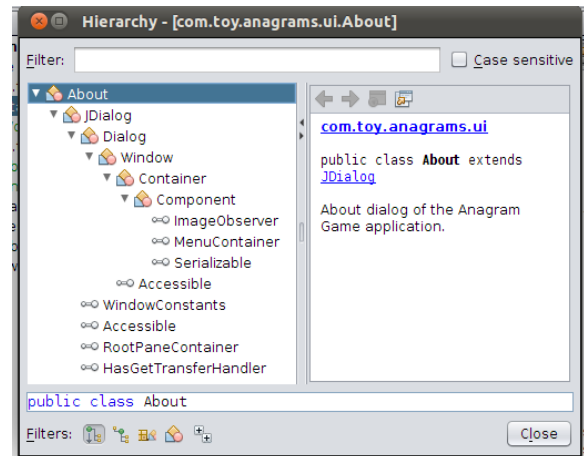
Press the same keys again and the commented lines will no longer be commented.

```
// public String getScrambledWord(int idx) {
//     return SCRAMBLED_WORD_LIST[idx];
// }
```

6. **Inspect members/hierarchy** – Both the members of the current class as well as its hierarchy can be displayed and then filtered. Press Alt-F12 and the ancestors of the current file are shown.

On the other hand, if you press Ctrl-F12, the current file's members are displayed, as shown here:

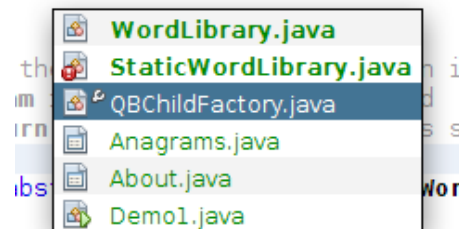
Source URL: <http://netbeans.dzone.com/news/10-handly-editor-shortcuts-netbeans-ide-60>



7. **Switch between documents** – When you have multiple documents open at the same time, press Ctrl and Tab and a small popup appears.

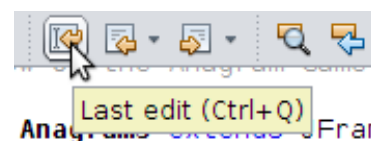
You can scroll through the popup, which displays all the open documents, and then choose the document that you want to open.

abstract String getWord(int idx



8. **Jump to last edit** – Often, you'll recently have coded in one Java file, while you currently find yourself in a completely different file. How do you get back to the place where you were last editing your code? That is likely to be the place where you want to get back to so that you can do some more editing.

Whenever you click Ctrl-Q, the last edited document is found and the cursor lands on the line where the last edit took place. Alternatively, you can click the button in the top left corner of the Source Editor.



9. **Bookmarks** – When you press Ctrl-Shift-M, the current line is “bookmarked.” This means that you can later quickly cycle back/forward to it (with Ctrl-Shift-Period and Ctrl-Shift-Comma). The bookmarked line gets a small icon in the left sidebar until you press Ctrl-Shift-M on the line again to remove the bookmark.

```

48 private int wordIdx = 0;
50 private WordLibrary wordLibrary;
    
```

10. **Highlight exit points** – Place the cursor on the return type and you will immediately see all exit points highlighted.

```

public boolean accept(File f){
    if (showHidden) {
        return true;
    } else {
        return false;
    }
}
    
```

RESOURCES

NetBeans DZone Community	http://netbeans.dzone.com/
NetBeans Weekly Newsletter:	http://www.netbeans.org/community/news/newsletter/latest.html
NetBeans Blogs	http://planet.netbeans.org/
NetBeans and Project Coin at JDK 7 Launch Event	http://www.oracle.com/us/corporate/events/java7/index.html

Thanks to the following people who kindly gave of their time and expertise in reviewing this refcard: Adam Bien, Tonny Kohar, Varun Nischal, Kristian Rink, Tom Wheeler, Tushar Joshi, and Joe Darcy.

ABOUT THE AUTHOR



Geertjan Wielenga is a Principal Product Manager at Oracle, focused on NetBeans. He has been part of the NetBeans team, as product manager, evangelist, and technical author, for the past 7 years. He lives in Amsterdam, in the Netherlands.

RECOMMENDED BOOK



The Well-Grounded Java Developer is a unique guide written for developers with a solid grasp of Java fundamentals. It provides a fresh, practical look at new Java 7 features along with the array of ancillary technologies that a working developer will use in building the next generation of business software.

Browse our collection of over 100 Free Cheat Sheets



Free PDF

Upcoming Refcardz

- Continuous Delivery
- CSS3
- NoSQL
- Android Application Development



DZone communities deliver over 6 million pages each month to more than 3.3 million software developers, architects and decision makers. DZone offers something for everyone, including news, tutorials, cheat sheets, blogs, feature articles, source code and more. **“DZone is a developer’s dream,”** says PC Magazine.

Copyright © 2011 DZone, Inc. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by means electronic, mechanical, photocopying, or otherwise, without prior written permission of the publisher.

DZone, Inc.
 140 Preston Executive Dr.
 Suite 100
 Cary, NC 27513
 888.678.0399
 919.678.0300
Refcardz Feedback Welcome
refcardz@dzone.com
Sponsorship Opportunities
sales@dzone.com

ISBN-13: 978-1-936502-40-0
 ISBN-10: 1-936502-40-2

50795

9 781936 502400

\$7.95