DZone Refcardz

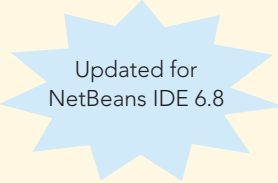## CONTENTS INCLUDE:

- About NetBeans IDE
- Java Editor Overview
- NetBeans IDE Java Quick Start Tutorial
- Keyboard Shortcuts and Code Templates
- Hot Tips and more...

Updated for NetBeans IDE 6.8

NetBeans Java Editor

By Geertjan Wielenga and Patrick Keegan

ABOUT NETBEANS IDE

The NetBeans IDE has seen adoption snowballing over the past years and now comes to you with deep Java EE 6 support in NetBeans IDE 6.8. You'll find this reference card helpful if you want to get as much out of the Java editor (and related tools) as its authors intended when creating it. It lists all the keyboard shortcuts in carefully thought out categories and it provides a thorough exposition of optimal handling of Java code in the editor, covering viewing, navigation, source handling, and refactoring. Go to http://www.netbeans.org/downloads/ to get NetBeans IDE.

JAVA EDITOR OVERVIEW
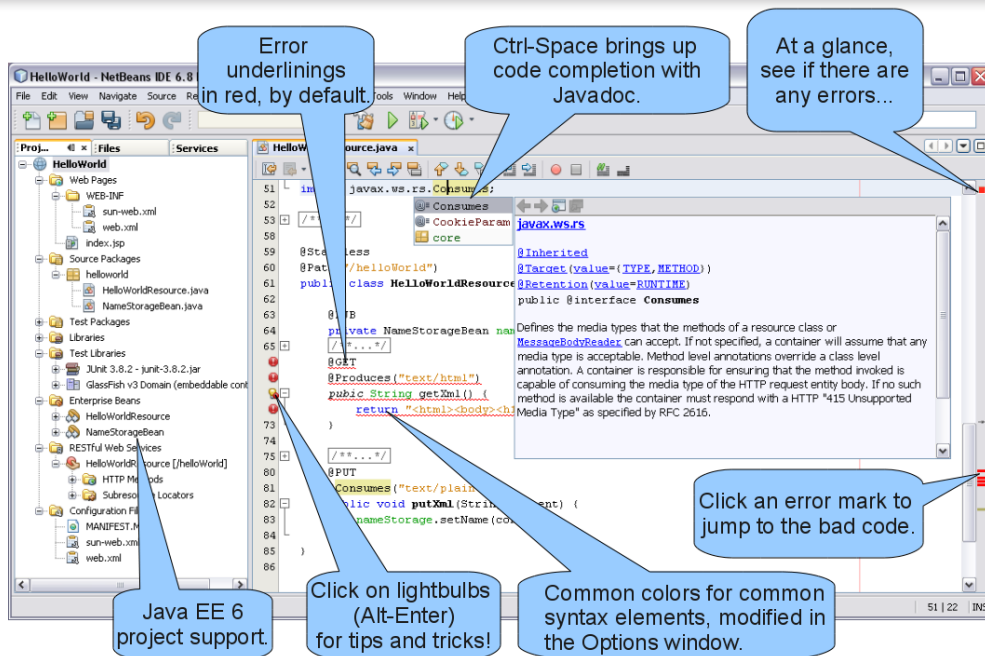
WHAT'S NEW FOR JAVA IN NETBEANS IDE 6.8

The following features are new in NetBeans IDE 6.8 and can be of particular use to you in the context of creating Java-based applications

Java EE 6 Support	The IDE provides tools for developing applications complying with the new Java Enterprise Edition 6, specifically web, EJB, web service, and JPA support. The Java EE 6 compliant server bundled with the IDE is GlassFish v3.
Facelets Editor	For Facelets (JavaServer Faces 2.0), a complete editor with coloring, code completion, namespace support, component palette, code folding, error checking, and code templates is provided.
Maven	Java EE 6 archetype support, new archetype catalogs, Groovy & Scala support, dependency exclusion support in visual dependency graph, as well as several other performance and usability improvements.
Other Related Enhancements	JavaFX editor, JIRA support, instant messaging, new UI for Kenai projects, and code completion enhancements in SQL editor.

NetBeans IDE 6.8 is the **first** IDE to provide complete support for the Java EE 6 specification and Sun Glassfish Enterprise Server v3. It is the tool of choice for top development languages, with support for Java, PHP 5.3 and the Symfony Framework, JavaFX, C/C++, and more. Get tomorrow today, download the NetBeans 6.8 IDE now.

NETBEANS IDE JAVA QUICK START TUTORIAL

This tutorial provides a very simple and quick introduction to the NetBeans IDE workflow by walking you through the creation of a simple "Hello World" Java console application.

1. Start NetBeans IDE. In the IDE, choose **File > New Project (Ctrl-Shift-N)**.
2. In the New Project wizard, expand the Java category and select Java Application. Then click Next.
3. In the Name and Location page of the wizard, type **"HelloWorld"** in Project Name. Then click Finish.
4. Because you have left the Create Main Class checkbox selected in the New Project wizard, the IDE has created a skeleton class for you. You can add the **"Hello World!"** message to the skeleton code by replacing the line:

   // TODO code application logic here

   with the line:

   System.out.println("Hello World!");

5. From the IDE's menu bar, choose **Run > Run Main Project (F6)**. The Output window should show you the **"Hello World!"** message.

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

Coding in Java

Alt-Insert	Generate code
Ctrl-Shift-I	Fix all class imports
Alt-Shift-I	Fix selected class's import
Alt-Shift-F	Format selection
Alt-Shift Left/Right/Up/Down	Shift lines left/right/up/down
Ctrl-Shift-Up/Down	Copy lines up/down
Ctrl/Alt-F12	Inspect members/hierarchy
Ctrl-/	Add/remove comment lines
Ctrl-E	Delete current line

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 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="" />
jspi	<jsp:include page="|"/>
jspp	<jsp:plugin type="|" code="" codebase=""></jsp:plugin>
jsps	<jsp:setProperty name="|" property=""/>
jspu	<jsp:useBean id="I" type=""/>

oup	out.print("|")
oupl	out.println("|")
pcg	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("|")
tglb	<%@taglib uri="|"%>

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. Firstly, the Options window on the Mac is found under **NetBeans > Preferences**.
Secondly, 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

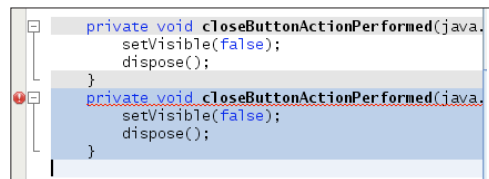| Keys	Action
Cmd-Z	Undo. Reverses a series of editor actions one at a time (excluding Save).
Cmd-Y	Redo. Reverses a series of Undo commands one at a time.
Cmd-X	Cut. Deletes the current selection and places it on the clipboard.
Cmd-C	Copy. Copies the current selection to the clipboard.
Cmd-V	Paste. Pastes the contents of the clipbard 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.

To Change Default Settings:

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.

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, as below:
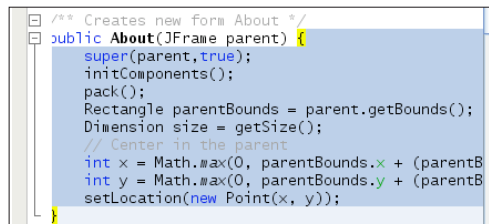
```
private void closeButtonActionPerformed(java.
    setVisible(false);
    dispose();
}
private void closeButtonActionPerformed(java.
    setVisible(false);
    dispose();
}
```

2. **Capture inner/outer syntactic element.** Each time you press Alt-Shift-Period, the selection expands to a successively wider syntactic element.
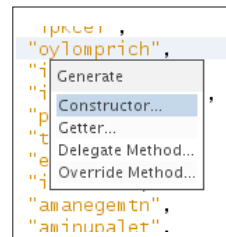
   For example, below one statement was selected, the key combination Alt-Shift-Period was pressed, and then the complete content of the surrounding block statement was shown to be selected.

   The selection expands from the current statement to sur rounding 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 + (parentB
    int y = Math.max(0, parentBounds.y + (parentB
    setLocation(new Point(x, y));
}
```
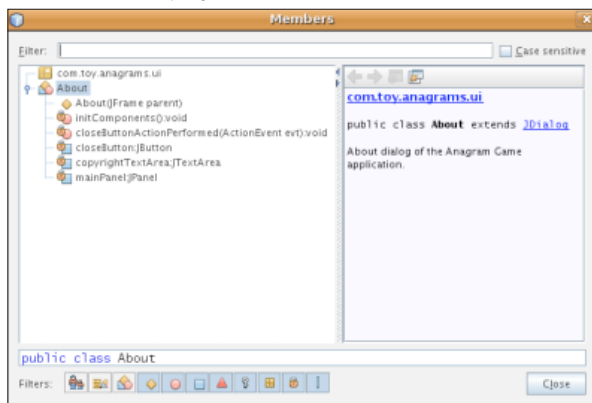
3. **Generate code skeletons.** Whenever you want to generate commonly needed pieces of code, such as constructors, getters, and setters, simply click Alt-Insert, and a small popup appears with a list of items from which you can select:

```
"oylomprich",
  Generate
  Constructor...
  Getter...
  Delegate Method...
  Override Method...
"amanegemtn",
"aminupalet",
```

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, as shown below. 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.get
    Dimension size = getSize();
    // Center in the parent
```

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

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

```
@return word at that index in its scrambl
 */
//    public static String getScrambledWord(int
//        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:

FYI   **Source URL:**
http://netbeans.dzone.com/news/10-handy-editor-shortcuts-netbeans-ide-60

---

7. Switch between documents. When you have multiple documents open at the same time, press Ctrl and Tab, and then 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:
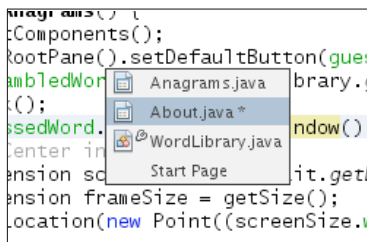
8. Jump to last edit. Often, you find yourself in a situation where you have edited in one document, while you currently find yourself in a completely different document. How do you get back to the place where you were last editing your code? That is likely to be the place where you want to get back to, in order to do some more editing. Now, 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 shown below, in the top left corner of the Source Editor:
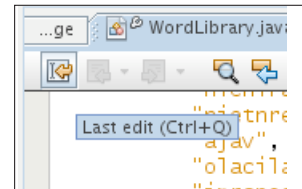
9. Bookmarks. When you press Ctrl-Shift-M, the current line is "bookmarked". What this means is that you can later quickly cycle back/forward to it (with Ctrl-Shift-Period and Ctrl-Shift-Comma). The bookmarked line gets a small icon in the left sidebar, as shown below, until you press Ctrl-Shift-M on the line again, to remove the bookmark:
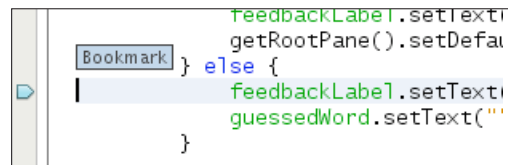
10. Highlight exit points. Place the cursor on the return type and you will immediately see all exit points highlighted:

```
@Override
public boolean accept(File f) {
    if (f.isDirectory()) {
        return true;
    }
    String fileName = f.getName();
    int i = fileName.lastIndexOf('.');
    if ((i > 0) && (i < (fileName.leng
        String fileExt = fileName.subs
        if ("txt".equalsIgnoreCase(fil
            return true;
        }
    }
    return false;
}
```

QUICK OPTIONS WINDOW OVERVIEW

The Options window lets you customize NetBeans IDE in a number of ways. Most people don't know how much can be customized there.

General	Sets the IDE-wide browser and the proxy settings.	▪ Web Browser ▪ Proxy Settings
Editor	Sets the editor-specific options, specifically those relating to code folding, code completion, camel case behavior, indentation, code templates, and macros.	▪ Code Folding ▪ Code Completion ▪ Camel Case Behavior ▪ Indentation ▪ Code Templates ▪ Macros
Fonts & Colors	Sets the fonts and colors for syntax, highlighting, annotations, and diff viewer.	▪ Syntax ▪ Highlighting ▪ Annotations ▪ Diff

Keymap	Sets the keyboard profile to be used throughout the IDE. By default, profiles are provided for NetBeans, Eclipse, and Emacs. A legacy profile is also provided, for NetBeans 5.5 keyboard shortcuts, which were radically rewritten in NetBeans IDE 6.0.	▪ NetBeans Profile ▪ Eclipse Profile ▪ Idea Profile ▪ Emacs Profile ▪ NetBeans 5.5 Profile
Miscellaneous	Sets the options for Ant processing, appearance, diffing, files, the Matisse GUI Builder, issue tracking, Java Debugger, JavaScript, Maven, Profiler, ToDo Tasks, and Versioning.	▪ Ant ▪ Appearance ▪ Diff ▪ Files ▪ GUI Builder ▪ Issue Tracking ▪ Java Debugger ▪ JavaScript ▪ Maven ▪ Profiler ▪ Tasks ▪ Versioning

RESOURCES

Resource	URL
NetBeans DZone Community	http://netbeans.dzone.com/
NetBeans Tutorials	http://www.netbeans.org/kb/index.html
NetBeans Video Tutorials	http://www.netbeans.org/kb/60/screencasts.html
NetBeans Blogs	http://planetnetbeans.org/
NetBeans TV	http://netbeans.tv/
NetBeans Weekly Newsletter:	http://www.netbeans.org/community/news/newsletter/latest.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, and Tom Wheeler.

ABOUT THE AUTHORS

Geertjan Wielenga is the NetBeans technical writer responsible for documentation related to the NetBeans Java editor. He is co-author of the book *Rich Client Programming: Plugging into the NetBeans Platform*. He is known for his popular blog at **http://blogs.sun.com/geertjan**, as well as for his role as a Zone Leader at **Javalobby**.

Patrick Keegan has been writing about the NetBeans IDE for over 9 years. In addition to writing help and tutorials, he is co-author of the *NetBeans IDE Field Guide* and has contributed to other books on NetBeans and Java.

RECOMMENDED BOOKS

Pro NetBeans IDE 6 Rich Client Platform Edition focuses on the new features of NetBeans 6 as well as what has changed since NetBeans 5.5, empowering you to be a more effective and productive developer.
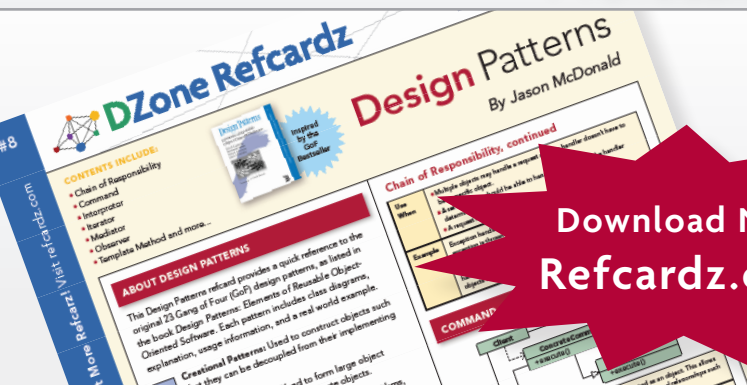
The Definitive Guide to NetBeans™ Platform is a thorough and definitive introduction to the NetBeans Platform, covering all its major APIs in detail, with relevant code examples used throughout.

BUY NOW
books.dzone.com/books/pronetbeans
books.dzone.com/books/definitive-guide-netbeans

DZone

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, cheatsheets, blogs, feature articles, source code and more. **"DZone is a developer's dream,"** says PC Magazine.

ISBN-13: 978-1-934238-60-8
ISBN-10: 1-934238-60-0

50795

9 781934 238608

$7.95

Version 1.0