

# Highlights of NetBeans IDE 10 Keyboard Shortcuts & Code Templates

## Finding, Searching, and Replacing

<b>^ \ G</b>	Search Word at Insert Point
<b>⌘ G</b>	Find Next / Previous in File
<b>⌘ R</b>	Find and Replace in File
<b>^ F7</b>	Find Usages
<b>⌘ ⌘ F / H</b>	Find / Replace in Projects
<b>^ ⌘ U</b>	Find Usages Results
<b>^ ⌘ H</b>	Toggle Search Result Highlights
<b>^ R</b>	Rename
<b>⌘ U, then U</b>	Convert Selection to Uppercase
<b>⌘ U, then L</b>	Convert Selection to Lowercase
<b>⌘ U, then S</b>	Toggle Case of Selection
<b>⌘ ⌘ V</b>	Paste Formatted
<b>⌘ ⌘ D</b>	Show Clipboard History
<b>⌘ I</b>	Jump to Quick Search Field
<b>^ ⌘ L</b>	Copy File Path

## Navigating through Source Code

<b>⌘ O / ^ ⌘ O</b>	Go to Type / File
<b>⌘ ⌘ T</b>	Go to JUnit Test
<b>⌘ ⌘ B</b>	Go to Source
<b>⌘ B</b>	Go to Declaration
<b>^ G</b>	Go to Line
<b>⌘ ⌘ M</b>	Toggle Bookmark
<b>⌘ ⌘ Period / Comma</b>	Next / Previous Bookmark
<b>⌘ ⌘ 1 / 2</b>	Select in Projects / Files
<b>⌘ [</b>	Move Caret to Matching Bracket
<b>^ Q</b>	Go To Last Edit Location
<b>^ ⌘ Period / Comma</b>	Select Next / Previous Element

## Coding in Java

<b>^ I</b>	Generate Code
<b>⌘ ⌘ I</b>	Fix Imports

<b>^ ⌘ I</b>	Fix Import for Selected Class
<b>^ ⌘ F</b>	Format Selection
<b>^ ⌘ ← / → / ↑ / ↓</b>	Select Left / Right / Up / Down
<b>^ ⌘ R</b>	Rectangular Selection (Toggle)
<b>^ ⌘ ↑ / ↓</b>	Copy Selection Up / Down
<b>⌘ F12</b>	Inspect Members / Hierarchy
<b>⌘ Shift C / ⌘ /</b>	Add / Remove Comment Lines
<b>⌘ E</b>	Delete Current Line

## Coding in C/C++

<b>^ ⌘ G</b>	Go to Declaration
<b>⌘ F9</b>	Evaluate Expression

## Compiling, Testing, and Running

<b>⌘ ⌘ U</b>	Create Unit Test
<b>⌘ ⌘ T</b>	Go to Unit Test
<b>^ F6 / ⌘ F6</b>	Run Unit Test on Project / File
<b>F6 / ⌘ F6</b>	Run Main Project / File
<b>⌘ ⌘ F6</b>	Debug Test File

## Opening and Toggling between Views

<b>⌘ `</b>	Go to Previous Document
<b>⌘ Esc</b>	Maximize / Minimize Window (Toggle)
<b>⌘ W</b>	Close Currently Selected Window
<b>⌘ ⌘ F4</b>	Close all Windows
<b>^ ⌘ D</b>	Undock / Redock Window (Toggle)
<b>⌘ ⌘ T</b>	Reopen Recently Closed File
<b>⌘ Mouse Wheel Up / Down</b>	Zoom Text In / Out
<b>⌘ ⌘ S</b>	Toggle Inspect Mode

## Debugging

<b>⌘ ⌘ F5</b>	Debug Selected File
<b>⌘ ⌘ F6</b>	Debug Test File
<b>F5</b>	Continue Debugger Session
<b>⌘ F5</b>	Finish Debugger Session
<b>F4</b>	Run to Cursor Location in File
<b>F7 / F8</b>	Step Into / Over
<b>⌘ F7</b>	Step Out
<b>^ ↑ / ↓</b>	Go to Called / Calling Method
<b>⌘ F7</b>	Evaluate Expression
<b>⌘ F8</b>	Toggle Breakpoint
<b>⌘ ⌘ F8</b>	New Breakpoint
<b>⌘ ⌘ F7</b>	New Watch

## Notes:

- ^ represents ctrl
- Press fn to use function keys (e.g., F5, F6, F8)

## Java Editor Code Templates

When typing in the Source Editor, generate the text in the right-column below by typing the abbreviation that is listed in the left-column and then pressing Tab.

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
as	assert true;
bcom	/**/

# Highlights of NetBeans IDE 10 Keyboard Shortcuts & Code Templates

bo	boolean
br	break;
ca	catch (
cl	class
cn	continue
db	double
df	default:
dowhile	do { } while ( <i>condition</i> );
eq	equals
ex	extends
fa	false
fcom	// <editor-fold></editor-fold>
fi	final
fl	float
forc	for (Iterator <i>it</i> = <i>collection.iterator()</i> ; <i>it.hasNext()</i> ;) { <i>Object elem</i> = ( <i>Object</i> ) <i>it.next()</i> ; }
fore	for ( <i>Object elem</i> : <i>iterable</i> ) { }
fori	for (int <i>i</i> = 0; <i>i</i> < <i>arr.length</i> ; <i>i</i> ++) {}
forl	for (int <i>i</i> = 0; <i>i</i> < <i>lst.size()</i> ; <i>i</i> ++) { <i>Object object</i> = <i>lst.get(i)</i> ; }
forst	for (StringTokenizer <i>st</i> = new StringTokenizer(""); <i>st.hasMoreTokens()</i> ;) }
forv	for (int <i>i</i> = 0; <i>i</i> < <i>vct.size()</i> ; <i>i</i> ++) { <i>Object object</i> = <i>vct.elementAt(i)</i> ;
fy	finally{ }
ie	interface
ifelse	if ( <i>condition</i> ){}else { }
iff	if (exp) {}
im	implements
inst	if ( <i>exp instanceof Object</i> ) { <i>Object obj</i> = ( <i>Object</i> ) <i>exp</i> ;
iof	instanceof
ir	import
le	length
na	native
newo	<i>Object name</i> = new <i>Object(args)</i> ;

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
runn	Runnable runnable = new Runnable() { public void run() {}};
serr	System.err.println ("");
sh	short
sout	System.out.println ("");
soutv	System.out.println(" <i>Object</i> = " + <i>Object</i> );
st	static
su	super
sw	switch (var) { case val: break; default: throw new AssertionError();}
sy	synchronized
tds	Thread.dumpStack();
th	throws
tr	transient
trycatch	try {} catch (Exception <i>e</i> ) {}
tw	throw
vo	volatile
wh	while (
whileit	while ( <i>it.hasNext()</i> ) { <i>Object elem</i> = ( <i>Object</i> ) <i>it.next()</i> ;
whilen	while ( <i>en.hasMoreElements()</i> ) { <i>Object elem</i> = ( <i>Object</i> ) <i>en.nextElement()</i> ;
whilexp	while (exp) {}
<b>JavaScript Editor Code Templates</b>	
al	window.alert( <i>{message}</i> );
br	<i>{no-indent}</i> break;
ca	<i>{no-indent}</i> case

	catch ( <i>{exception}</i> ) { <i>{cursor}</i> }
cond	window.console.debug( <i>{debug}</i> );
coni	window.console.info( <i>{info}</i> );
conl	window.console.log( <i>{log}</i> );
conw	window.console.warn( <i>{warn}</i> );
ct	<i>{no-indent}</i> catch
do	<i>{no-indent}</i> document
dw	document.write( <i>{message}</i> );
	Else {
else	<i>{cursor}</i> }
eq	<i>{no-indent}</i> equals
fa	<i>{no-indent}</i> false
forin	for (var <i>{item}</i> in <i>{object}</i> ) { <i>{selection}{cursor}</i> }
	function <i>{name}</i> ( <i>{parameters}</i> ) { <i>{selection}{cursor}</i> }
fun	
hi	<i>{no-indent}</i> window.history
	if ( <i>{expr}</i> ) {
if	<i>{selection}{cursor}</i> }
iof	<i>{no-indent}</i> instanceof
json	{ <i>"{field}": "{value}"</i> }
lo	<i>{no-indent}</i> window.location
new	<i>{no-indent}</i> var <i>{name}</i> = new <i>{type}</i> ( <i>{arguments}</i> );
	<i>{propertyName}</i> : function( <i>{parameters}</i> ) {
prf	<i>{cursor}</i> },
re	<i>{no-indent}</i> return
var	<i>{no-indent}</i> var <i>{name}</i> = <i>{value}</i> ;
win	<i>{no-indent}</i> window