NetBeans IDE Cheatsheet

A comprehensive cheat sheet for the NetBeans IDE, covering essential features, shortcuts, debugging, and project management tips for efficient development.

General Usage & Shortcuts

Basic Navigation

Ctrl + Shift + N	New Project
Ctrl + N	New File
Ctrl + O	Open File
Ctrl + Shift + O	Open Project
Ctrl + S	Save
Ctrl + Shift + S	Save All
Ctrl + W	Close Current Tab
Ctrl + Shift + W	Close All Tabs

Editing Shortcuts

Ctrl + X	Cut Line
Ctrl + C	Copy Line
Ctrl + V	Paste
Ctrl + Z	Undo
Ctrl + Y	Redo
Ctrl + D	Duplicate Line
Alt + Shift + Up/Down	Move Line Up/Down
Ctrl + Shift + Up/Down	Select Multiple Lines

Code Completion & Generation

Ctrl + Space	Code Completion
Alt + Insert	Generate Code (Getters, Setters, Constructors, etc.)
Ctrl + I	Fix Imports
Ctrl + Shift + I	Reformat Code
Ctrl + /	Comment/Uncomment Line
Alt + Shift + F	Format Code

Debugging

Debugging Basics

Ctrl + F5	Start Debugging
Shift + F5	Continue Debugging
Ctrl + Shift + F5	Stop Debugging
F7	Step Into
F8	Step Over
Ctrl + F7	Step Out
Ctrl + Shift + B	Toggle Breakpoint
Ctrl + F9	Evaluate Expression

Breakpoints & Watches

Set breakpoints by clicking in the left margin of the editor next to the line number. Use conditions on breakpoints to pause execution only when certain conditions are met.

Add variables to the 'Watches' window (Window > Debugging > Watches) to monitor their values during debugging.

Use the 'Call Stack' window (Window > Debugging > Call Stack) to see the sequence of method calls that led to the current point in the execution.

Use conditional breakpoints to stop the execution only when a certain condition is met. Right-click a breakpoint and select 'Properties' to set a condition.

Debugging Windows

Variables Window: Displays the current values of variables in the current scope.

Watches Window: Allows you to monitor specific expressions or variables.

Call Stack Window: Shows the sequence of method calls.

Threads Window: Manages and monitors threads in a multithreaded application.

Project Management

Project Structure

NetBeans organizes projects into logical units, making it easier to manage large codebases. Key directories include `src` for source code, `lib` for libraries, and `build` for compiled output.

Use the 'Projects' window (Ctrl + 1) to view and manage project files and dependencies.

Right-click on a project in the 'Projects' window to access options like 'Clean and Build', 'Run', 'Debug', and 'Properties'.

Build & Run

Ctrl + Shift + B	Clean and Build Project
F6	Run Project
Shift + F6	Run File
Ctrl + F6	Debug Project

Version Control

NetBeans provides built-in support for Git, Mercurial, and Subversion. Use the 'Team' menu to access version control features.

Right-click on a project or file to commit, update, push, and perform other version control operations.

Use the 'Diff' window to compare changes between different versions of a file.

Resolve conflicts by using the merge tools available in NetBeans.

Advanced Features

Refactoring

NetBeans offers powerful refactoring tools to improve code quality and maintainability. Access refactoring options by right-clicking on a code element (e.g., variable, method, class) and selecting 'Refactor'.

Rename: Safely rename variables, methods, and classes throughout your project.

Move: Move classes to different packages.

Extract Method: Create a new method from a selected block of code.

Inline: Replace a method call with the method's body.

Code Templates

NetBeans allows you to define and use code templates (also known as live templates) to quickly insert commonly used code snippets. Access code templates via 'Tools > Options > Editor > Code Templates'.

Type the abbreviation for a template and press Tab to expand it. For example, type `sout` and press Tab to insert `System.out.println();`.

Customize existing templates or create new ones to suit your coding style.

Plugins

Extend NetBeans functionality by installing plugins. Access the Plugin Manager via 'Tools > Plugins'.

Search for plugins in the NetBeans Plugin Portal or install them from local files.

Popular plugins include those for additional language support, version control integration, and code analysis tools.