Editor. Smart Keys

Use this page to enable or disable specific smart keys and to define which actions you want to be invoked automatically.

Item	Description
Home	When this check box is selected, on pressing Home, the caret is positioned at the first non-space character of the current line. Pressing Home subsequently moves the caret from the *Smart Home position* to the first column and back.
End (on blank line)	When this check box is selected, on pressing End in an empty line, the caret is positioned with the indent, which IntelliJ IDEA assumes to be reasonable in the current code point (indentation is based on the current Code Style Settings).
Insert pair bracket	Select this check box to have IntelliJ IDEA automatically add a closing round or square bracket for each typed opening round or square bracket, respectively.
Insert pair quote	Select this check box to have IntelliJ IDEA automatically add a closing single or double quote for each typed opening single or double quote, respectively.

Item	Description
Use 'CamelHumps' words	Select this check box to have IntelliJ IDEA discern separate words within CamelHump names. Words within a name should start with a capital letter or an underscore. This option impacts some editor actions, for example: ■ *Caret Move* (`Ctrl+Right`/`Ctrl+Left`) ■ *Caret Move with Selection* (`Ctrl+Shift+Right`/`Ctrl+Shift+Left`) ■ *Select Word at Caret* (`Ctrl+W`) ■ *Delete to Word Start/End* (`Ctrl+Backspace` and `Ctrl+Delete` respectively) ■ *Double-clicking* The following editor actions work in a mode opposite to the one established by the **Use 'CamelHumps' words** setting: ■ *Move Caret to Previous Word in Different "CamelHumps" mode* ■ *Move Caret to Previous Word with Selection in Different "CamelHumps" mode* ■ *Move Caret to Next Word in Different "CamelHumps" mode* ■ *Move Caret to Next Word with Selection in Different "CamelHumps" mode* ■ *Delete to Word End in Different "CamelHumps" mode* ■ *Delete to Word Start in Different "CamelHumps" mode* For example, If the check box is selected, the action *Move Caret to Next Word in Different "CamelHumps" mode* moves the caret to the end of word; if the check box is not selected, then the caret moves to the next CamelHump. These actions have no default keyboard shortcuts, and are not included in the menus. Thus, they can be invoked by pressing `Ctrl+Shift+A`: (Refer to the section Navigating to Action) You can bind them with the shortcuts of your choice as described in the section Configuring Keyboard Shortcuts.
Surround selection on typing quote or brace	If this check box is selected, the selected text on typing a quote, double-quote or brace, will be surrounded with these characters. If this check box is not selected, then the typed quotes, double-quotes or braces will replace the selection.
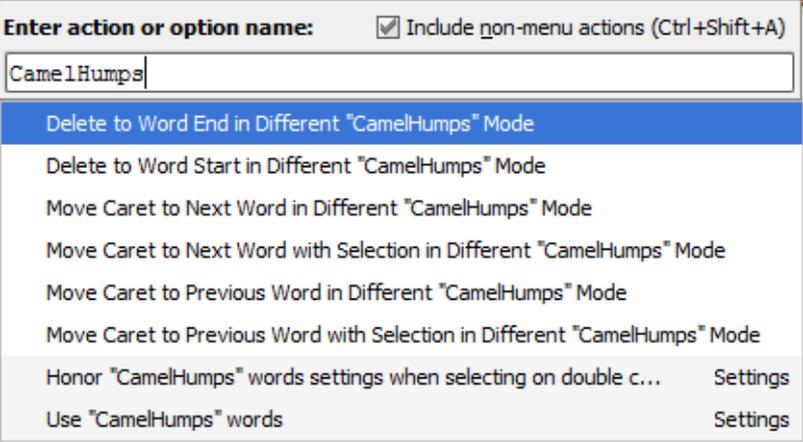
Item	Description

Enter	Use this area to define the actions to be invoked by pressing `Enter`.
	
	■ **Smart Indent** - select this check box to have IntelliJ IDEA add a new line and position the caret in it, with the indent that IntelliJ IDEA assumes to be reasonable in the current point of code (indentation is based on the current Code Style Settings).
	
	If the check box is cleared, upon pressing `Enter` in a blank line, IntelliJ IDEA adds a new line and positions the caret at the current non-space character column.
	
	■ **Insert pair '}'** - select this check box to have IntelliJ IDEA automatically position a closing brace `}` at the proper column when `Enter` is pressed in an empty line. In this case IntelliJ IDEA seeks backward for the nearest unclosed opening brace `{` and places the closing one at the corresponding indentation level.
	
	■ **Insert documentation comment stub** - this check box defines the behavior on pressing `Enter` after opening tag.
	
	  ■ If this check box is selected, IntelliJ IDEA generates a documentation comment stub. For the method comments, this stub contains the required tags (`@param` tags for each method parameter, `@return`, or `@throws`). Refer to Creating Documentation Comments for details.
	
	  ■ If this check box is not selected, only the closing tag is generated.

Reformat on paste	Use this drop-down list to specify how to place pasted code blocks. The available options are: ■ *None* - The pasted code is inserted at the caret location as plain text without any reformatting or indenting. ■ *Indent Block* - The pasted code block is positioned at the proper indentation level, according to the current Code Style Settings, but its inner structure is not changed. ■ *Indent Each Line*- Each line of the pasted code block is positioned at the proper indentation level, according to the current *Code Style Settings*. ■ *Reformat Block*- The pasted code block is reformatted according to the current *Code Style Settings*. This feature is applicable to lines that contain the trailing line feed characters.

Item	Description
Javadoc	In this area, define the behavior of the closing tags in Javadoc. ■ *Automatically insert closing tag* - select this check box to make IntelliJ IDEA insert closing tag automatically, after typing `>`. So doing, the caret rests inside the tag. For example, if you type `<b>`, the closing tag`</b>` will be generated automatically.
Insert pair '%>' on Enter in JSP	Select this check box to have IntelliJ IDEA automatically position the opening angle bracket `<` at the proper column when entered in an empty line in JSP code. In this case IntelliJ IDEA seeks backward for the nearest unclosed angle bracket and places a closing one `>` at the corresponding indentation level.
XML/HTML	In this area, define the actions to be invoked automatically when editing *XML* or *HTML* code. ■ *Automatically insert closing tag*: select this check box to have IntelliJ IDEA automatically insert a closing XML or HTML tag upon entering the corresponding opening one. ■ *Automatically insert required attributes*: select this check box to have IntelliJ IDEA display a template with all mandatory attributes of the typed tag. ■ *Automatically start attribute*: select this check box to have IntelliJ IDEA display a template with the first mandatory attribute of the typed tag.

See Also

Procedures:

■ Cutting, Copying and Pasting

■ Completing Statements

■ Creating Documentation Comments

Reference:

■ Editor. Code Completion

Web Resources:

■ Developer Community ⊡