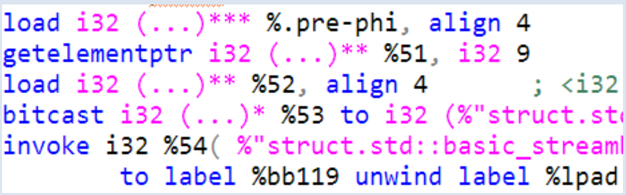
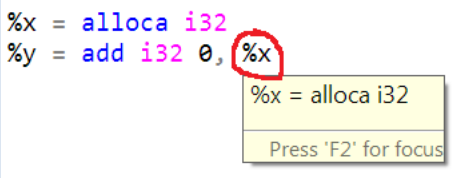
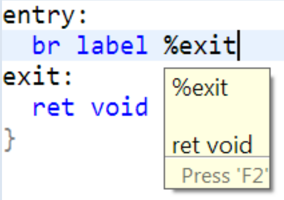
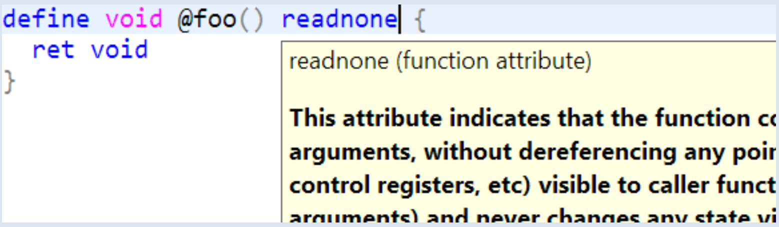
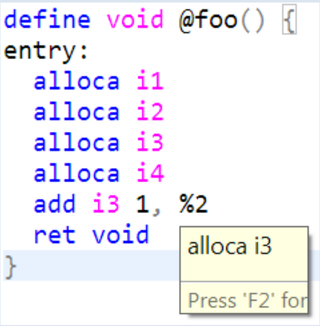
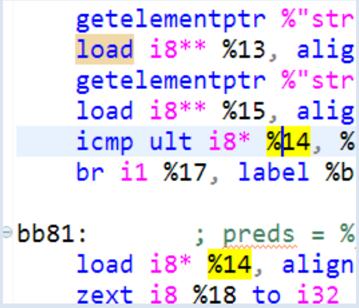
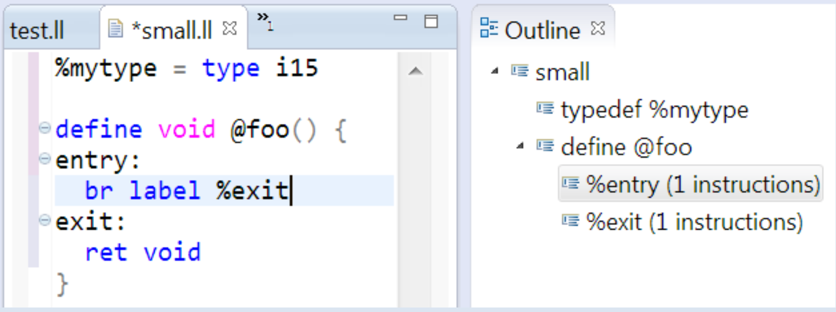
LLVM IR Editor Plugin for Eclipse: Features

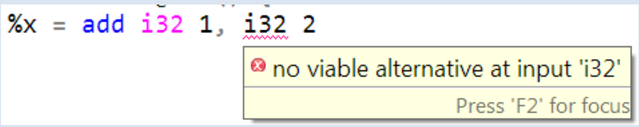
Syntax highlighting, which is aware of the complex type system	 load i32 (...)*** %.pre-phi, align 4 getelementptr i32 (...)** %51, i32 9 load i32 (...)** %52, align 4 ; <i32 bitcast i32 (...)* %53 to i32 (%"struct.st invoke i32 %54(%"struct.std::basic_streaml to label %bb119 unwind label %lpad
Hover support for values, including globals and user-defined types	 %x = alloca i32 %y = add i32 0, %x %x = alloca i32 Press 'F2' for focus Mouse cursor is over the circled %x. For the rest of the screenshots just assume the mouse is hovering over the element just above the popup box.
Hover support for basic blocks, showing the actual instructions in the block (or an excerpt, if the block is very large)	 entry: br label %exit exit: ret void } %exit ret void Press 'F2'

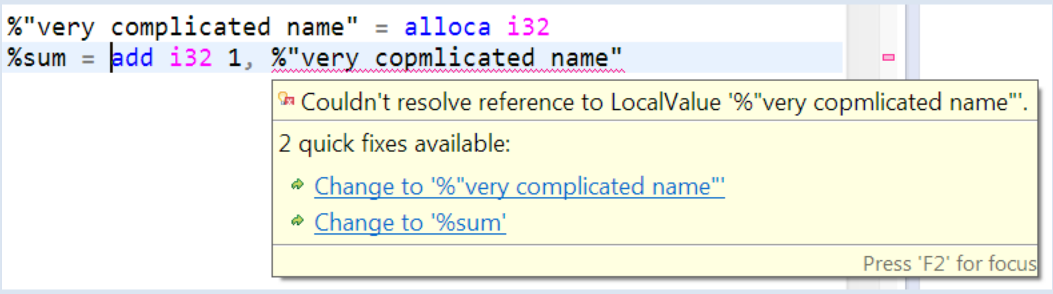
Built-in documentation on hover for (some) keywords	
	define void @foo() readnone {
	ret void
	}
	readnone (function attribute)
	This attribute indicates that the function c
	arguments, without dereferencing any poi
	control registers, etc) visible to caller funct
	arguments) and never changes any state vi

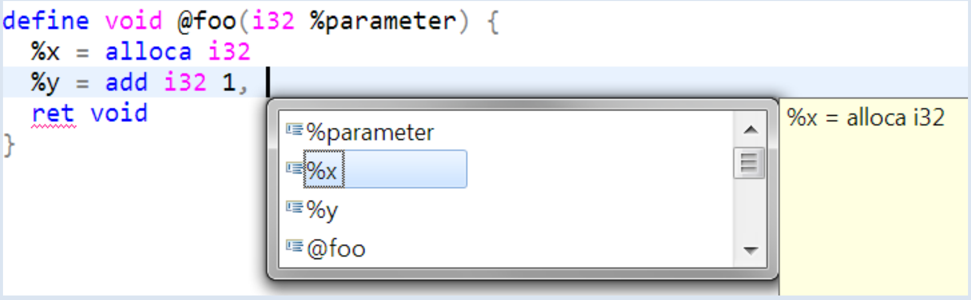
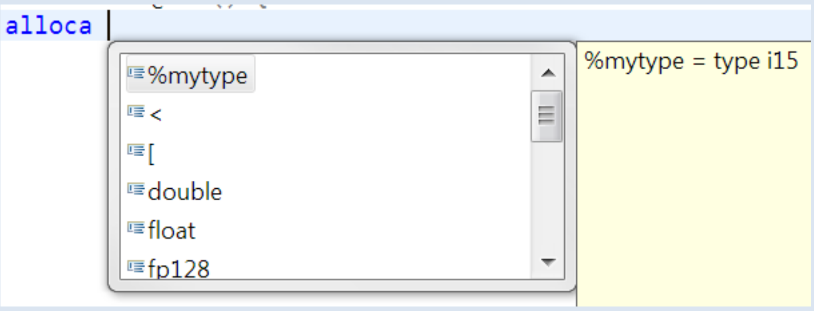
Inferring names of anonymous elements	
	define void @foo() {
	entry:
	alloca i1
	alloca i2
	alloca i3
	alloca i4
	add i3 1, %2
	ret void
	}
	alloca i3
	Press 'F2' for

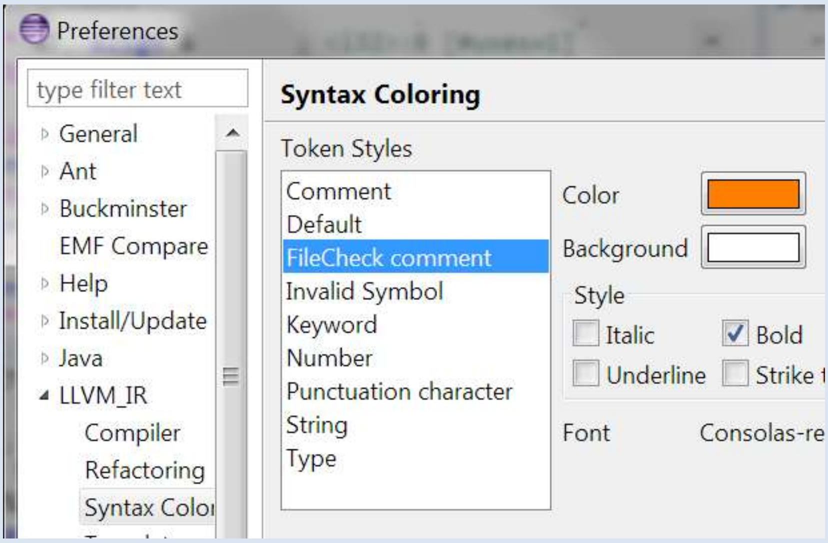
(Optional) Marking occurrences of element under cursor, with the option of marking definition and references in different colors	 getelementptr %"str load i8** %13, alig getelementptr %"str load i8** %15, alig icmp ult i8* %14, % br i1 %17, label %b bb81: ; preds = % load i8* %14, align zext i8 %18 to i32

A side window showing an outline of globals, functions and basic blocks in the file, and focusing on the active one	
	 test.ll
	*small.ll
	Outline
	`%mytype = type i15`
	`define void @foo() {`
	`entry:`
	`  br label %exit`
	`exit:`
	`  ret void`
	`}`
	 small
	typedef %mytype
	define @foo
	%entry (1 instructions)
	%exit (1 instructions)

Catching (some) syntax errors	 `%x = add i32 1, i32 2` no viable alternative at input 'i32' Press 'F2' for focus

"Quick fix" for elements	
	`%"very complicated name" = alloca i32`
	`%sum = add i32 1, %"very copmlicated name"`
	Couldn't resolve reference to LocalValue '%"very copmlicated name"'.
	2 quick fixes available:
	Change to '%"very complicated name"'
	Change to '%sum'
	Press 'F2' for focus

Content completion for values, types and basic blocks	
	`define void @foo(i32 %parameter) {`
	`  %x = alloca i32`
	`  %y = add i32 1,`
	`  ret void`
	`}`
	
	%x = alloca i32
	
	`alloca`
	
	%mytype = type i15

Configurable	 Preferences type filter text General Ant Buckminster EMF Compare Help Install/Update Java LLVM_IR Compiler Refactoring Syntax Color Syntax Coloring Token Styles Comment Default FileCheck comment Invalid Symbol Keyword Number Punctuation character String Type Color 🟧 Background □ Style □ Italic ☑ Bold □ Underline □ Strike t Font Consolas-re

… and other things that come “for free” with Eclipse, such as

- Spell check in comments
- Incremental and “under cursor” search options
- Shortcuts for commenting and uncommenting blocks of code
- Shortcuts for moving current line / selection up or down
- Shortcut for going to matching parenthesis