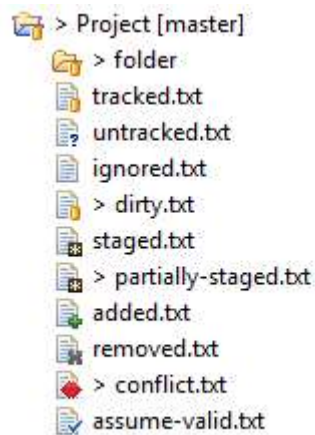**eGit Icon Decorations**
Icon decorations appear on the lower left corner of the icon label. They can be configured on the Preferences dialog under Team -> Git -> Label Decorations on the tab "Icon Decorations".

These are the default decorations:

- **dirty (folder)** - At least one file below the folder is dirty; that means that it has changes in the working tree that are neither in the index nor in the repository.
- **tracked** - The resource is known to the Git repository.
- **untracked** - The resource is not known to the Git repository.
- **ignored** - The resource is ignored by the Git team provider. Here only the preference settings under Team -> Ignored Resources and the "derived" flag are relevant. The .gitignore file is not taken into account.
- **dirty** - The resource has changes in the working tree that are neither in the index nor in the repository.
- **staged** - The resource has changes which are added to the index. Not that adding to the index is possible at the moment only on the commit dialog on the context menu of a resource.

- **partially-staged** - The resource has changes which are added to the index and additionally changes in the working tree that are neither in the index nor in the repository.
- **added** - The resource is not yet tracked by but added to the Git repository.
- **removed** - The resource is staged for removal from the Git repository.
- **conflict** - A merge conflict exists for the file.
- **assume-valid** - The resource has the "assume unchanged" flag. This means that Git stops checking the working tree files for possible modifications, so you need to manually unset the bit to tell Git when you change the working tree file. This setting can be switched on with the menu action Team->Assume unchanged (or on the command line with git update-index --assume-unchanged).

JDT Icons

Objects

	compilation unit (*.java file)
	Java file which is not on a build path
	class file
	generic file (unknown content type)
	unknown object
	Java scrapbook page (*.jpage file)
	Java scrapbook page (evaluation in progress)
	JAR description file
	JUnit test result file
	Java working set
	Java model
	library container
	JAR file with attached source
	JAR file without attached source
	class folder with attached source
	class folder without attached source
	source folder
	package
	empty package
	logical package
	empty logical package

	package only containing non Java resources
	package declaration
	import container
	import
	class (public)
	interface (public)
	enum type (public)
	annotation type (public)
	package visible class
	private class
	protected class
	default field (package visible)
	private field
	protected field
	public field
	default method (package visible)
	private method
	protected method
	public method

Object adornments

	marks project as Java project
	decorates files and folders if they are on the build path of their enclosing Java project
	decorates Java projects and working sets that contain build path errors
	this Java element causes an error
	this Java element causes a warning
	this Java element is deprecated
	constructor
	abstract member
	final member
	static member
	synchronized member
	native method
	transient field
	volatile field
	type with public static void main(String[] args)
	implements method from interface
	overrides method from super class
	type with focus in Type Hierarchy or Quick Outline/Hierarchy
	maximal expansion level in Call Hierarchy

	recursive call in Call Hierarchy
	compilation unit containing an abstract class as primary type
	compilation unit containing an interface as primary type
	compilation unit containing an enum as primary type
	compilation unit containing an annotation as primary type

Build path

	class path variable
	JAR with attached source
	JAR without attached source
	system library
	build path ordering
	inclusion filter
	exclusion filter
	access rules
	Javadoc location
	source attachment
	native library location
	output folder

Code assist

‹ ›	HTML tag
@	Javadoc tag
	local variable
	template
SWT	SWT template

Compare

	field
	method

Debugger

	debug launch
	run launch
	terminated run launch
	process
	terminated process
	debug target
	suspended debug target
	terminated debug target
	thread
	suspended thread
	stack frame
	running stack frame

	adornment that marks a stack frame that may be out of synch with the target VM as a result of an unsuccessful hot code replace
	adornment that marks a stack frame that is out of synch with the target VM as a result of an unsuccessful hot code replace
	inspected object or primitive value
	watch expression
	local variable
	monitor
	a monitor in contention
	a thread in contention for a monitor
	a monitor that is owned by a thread
	a thread that owns a monitor
	current instruction pointer (top of stack)
	current instruction pointer
	enabled line breakpoint
	disabled line breakpoint
	adornment that marks a breakpoint as skipped
	adornment that marks a line breakpoint as installed
	adornment that marks a breakpoint as conditional
	adornment that marks an entry method breakpoint

	adornment that marks an exit method breakpoint
	field access watchpoint
	field modification watchpoint
	field access and modification watchpoint
	adornment that marks a watchpoint as installed
	exception breakpoint
	runtime exception breakpoint
	disabled exception breakpoint
	adornment that marks an exception breakpoint as caught
	adornment that marks an exception breakpoint as uncaught
	adornment that marks an exception breakpoint as scoped
	adornment that marks an exception breakpoint as installed

Editor

	implements
	overrides
	quick assist available
	search match

JUnit

	test
	currently running test
	successful test
	failing test
	test throwing an exception
	ignored test
	test with an assumption failure
	test suite
	currently running test suite
	successfully completed test suite
	test suite with failing test
	test suite with exception throwing test
	caught exception
	stack frame element

NLS tools

	skipped NLS key
	translated NLS key
	untranslated NLS key

Quick fix

	quick fixable error
	quick fixable warning

✚	add
⤳	change
().	change cast
📷	fix multiple problems
⊞	move to another package
━	remove
⬜ˣ	remove import
⤳	rename
⬚	rename in file
🔧	surround with try/catch

Refactoring

⤴	general change
⤴	composite change
📄	text change
📄	file change
⊖	Stop error
⊗	Error
⚠	Warning
ℹ	Information
🔀	Change filter

Search

🔍	Java Search
🔖	search for declarations
🔖	search for references

Search - Occurrences in File

⇨	a general match
🔍	read access to local or field
✎	write access to local or field

Type Hierarchy view

ⓒ	type outside of selected package
ⓘ	interface outside of selected package