

# DeveloperManual DirStructure

## JMeter's Directory Structure

JMeter is organized by protocols and functionality. This is done so that developers can build new jars for a single protocol without having to build the entire application. We'll go into the details of building JMeter later in the tutorial. Since most of the jmeter developers use eclipse, the article will use eclipse directory as a reference point.

### Top Level Directories

- **bin** – contains the .bat and .sh files for starting Jmeter. It also contains ApacheJMeter.jar and properties file
- **build** – Created by build script, this is where various artifacts created during the build process are kept
- **dist** – Created by the build script, this is where distributions are generated
- **docs** – directory contains the JMeter documentation files
- **extras** – ant related extra files
- **lib** – contains the required jar files for Jmeter
- **src** – contains subdirectory for each protocol and component
- **test** – unit test related directory
- **xdocs** – Xml files for documentation. JMeter generates documentation from Xml.

### Lib Directory Structure

#### lib/

- **ext** – contains the core jar files for jmeter and the protocols. The reason for the separation is JMeter searches these jars for JMeter specific classes. Startup would be much slower if JMeter searched all the jars in the /lib directory.
- **opt** – contains optional jar files that JMeter can use for additional functionality. These jars are included when building and running JMeter, but they are not stored in CVS (usually due to licensing issues). Users can download optional jars separately and put here (they may have to create the directory).

### Src Directory Structure

#### src/

- **core** – the core code of JMeter including all core interfaces and abstract classes.
- **components** – contains non-protocol-specific components like visualizers, assertions, etc..
- **examples** – example sampler demonstrating how to use the new bean framework
- **functions** – standard functions used by all components
- **htmlparser** – a snapshot of [HtmlParser](#), donated by [HtmlParser](#) project on sourceforge
- **jorphan** – utility classes providing common utility functions
- **monitor** – Tomcat 5 monitor components
- **reports** – Report generating tool for processing test results
- **protocol** – contains the different protocols JMeter supports

### Protocol Directory Structure

#### src/protocol/

- **http** – components for load testing servers that use HTTP (web servers, soap services, xml-rpc services, etc)
- **ftp** – components for load testing ftp servers
- **java** – components for load testing java components
- **jdbc** – components for load testing database servers using jdbc
- **jndi** – components for load testing jndi (currently non-functional)
- **ldap** – components for load testing LDAP servers
- **mail** – components for load testing mail servers
- **tcp** – components for load testing TCP services
- **jms** – components for load testing JMS messaging services

As a general rule, all samplers related to HTTP will reside in "http" directory. The exception to the rule is the Tomcat5 monitor. It is separate, because the functionality of the monitor is slightly different than stress or functional testing. It may eventually be reorganized, but for now it is in its own directory. In terms of difficulty, writing visualizers is probably one of the harder plugins to write.