Index Next Prev

About

- Overview
- Changes
- Issues
- License
- Contributors

Download

- Download Releases
- Developer (Nightly) Builds

Documentation

- User Manual
- Javadocs
- Localisation (Translator's Guide)
- Building JMeter and Add-Ons
- JMeter Wiki
- FAQ (Wiki)

Tutorials (PDF format)

- Distributed Testing
- Recording Tests
- JUnit Sampler
- Access Log Sampler
- Extending JMeter

Community

- Get Involved
- Mailing Lists
- SVN Repositories

Foundation

- ASF
- Sponsorship
- Thanks

18.1 Samplers

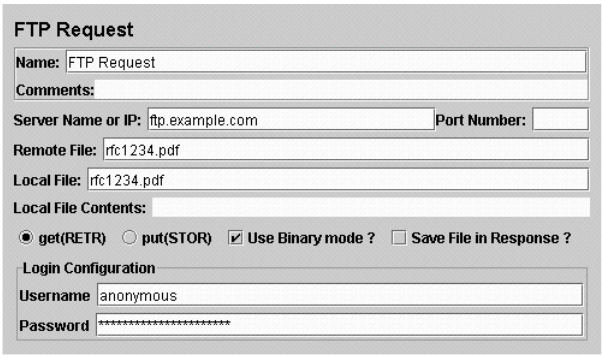
Samplers perform the actual work of JMeter. Each sampler (except Test Action) generates one or more sample results. The sa
various attributes (success/fail, elapsed time, data size etc) and can be viewed in the various listeners.

18.1.1 FTP Request

This controller lets you send an FTP "retrieve file" or "upload file" request to an FTP server. If you are going to send multiple
FTP server, consider using a FTP Request Defaults Configuration Element so you do not have to enter the same information f
Generative Controller. When downloading a file, it can be stored on disk (Local File) or in the Response Data, or both.

Latency is set to the time it takes to login (versions of JMeter after 2.3.1).

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this controller that is shown in the tree.
Server Name or IP	Domain name or IP address of the FTP server.
Port	Port to use. If this is >0, then this specific port is used, otherwise JMeter uses the default FTP port.
Remote File:	File to retrieve or name of destination file to upload.
Local File:	File to upload, or destination for downloads (defaults to remote file name).
Local File Contents:	Provides the contents for the upload, overrides the Local File property.
get(RETR) / put (STOR)	Whether to retrieve or upload a file.
Use Binary mode ?	Check this to use Binary mode (default Ascii)
Save File in Response ?	Whether to store contents of retrieved file in response data. If the mode is Ascii, then the contents will b visible in the Tree View Listener.
Username	FTP account username.
Password	FTP account password. N.B. This will be visible in the test plan.

See Also:

- Assertions
- FTP Request Defaults
- Building an FTP Test Plan

18.1.2 HTTP Request

This sampler lets you send an HTTP/HTTPS request to a web server. It also lets you control whether or not JMeter parses HT and other embedded resources and sends HTTP requests to retrieve them. The following types of embedded resource are retri

- images
- applets
- stylesheets
- external scripts
- frames
- background images (body, table, TD, TR)
- background sound

The default parser is htmlparser. This can be changed by using the property "htmlparser.classname" - see jmeter.properties for

If you are going to send multiple requests to the same web server, consider using an HTTP Request Defaults Configuration El have to enter the same information for each HTTP Request.

Or, instead of manually adding HTTP Requests, you may want to use JMeter's HTTP Proxy Server to create them. This can sa have a lot of HTTP requests or requests with many parameters.

There are two different screens for defining the samplers:

- AJP/1.3 Sampler - uses the Tomcat mod_jk protocol (allows testing of Tomcat in AJP mode without needing Apache ht does not support multiple file upload; only the first file will be used.
- HTTP Request - this has an implementation drop-down box, which selects the HTTP protocol implementation to be use
  - Java - uses the HTTP implementation provided by the JVM. This has some limitations in comparison with the Ht implementations - see below.
  - HTTPClient3.1 - uses Apache Commons HttpClient 3.1. This is no longer being developed, and support for this r future JMeter release.
  - HTTPClient4 - uses Apache HttpComponents HttpClient 4.x.

The Java HTTP implementation has some limitations:

- There is no control over how connections are re-used. When a connection is released by JMeter, it may or may not be re thread.
- The API is best suited to single-threaded usage - various settings (e.g. proxy) are defined via system properties, and the connections.
- There is a bug in the handling of HTTPS via a Proxy (the CONNECT is not handled correctly). See Java bugs 6226610
- It does not support virtual hosts.

Note: the FILE protocol is intended for testing puposes only. It is handled by the same code regardless of which HTTP Sampl

If the request requires server or proxy login authorization (i.e. where a browser would create a pop-up dialog box), you will al HTTP Authorization Manager Configuration Element. For normal logins (i.e. where the user enters login information in a for work out what the form submit button does, and create an HTTP request with the appropriate method (usually POST) and the from the form definition. If the page uses HTTP, you can use the JMeter Proxy to capture the login sequence.

In versions of JMeter up to 2.2, only a single SSL context was used for all threads and samplers. This did not generate the pro users. A separate SSL context is now used for each thread. To revert to the original behaviour, set the JMeter property:

```
https.sessioncontext.shared=true
```

JMeter defaults to the SSL protocol level TLS. If the server needs a different level, e.g. SSLv3, change the JMeter property, f

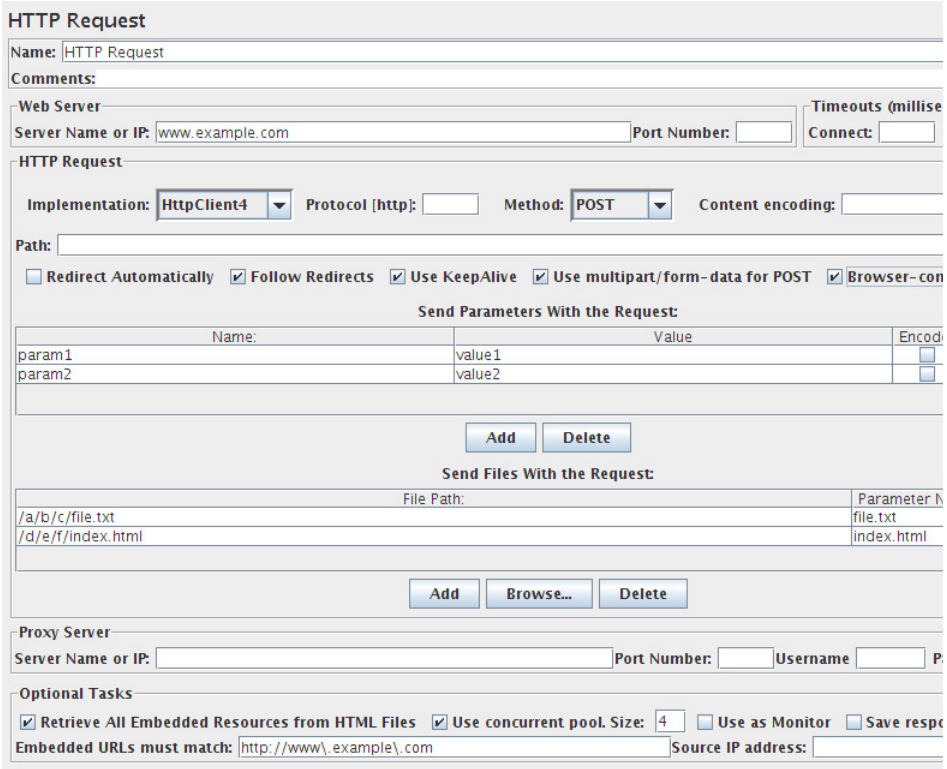
```
https.default.protocol=SSLv3
```

JMeter also allows one to enable additional protocols, by changing the property `https.socket.protocols` .

If the request uses cookies, then you will also need an HTTP Cookie Manager . You can add either of these elements to the T HTTP Request. If you have more than one HTTP Request that needs authorizations or cookies, then add the elements to the T way, all HTTP Request controllers will share the same Authorization Manager and Cookie Manager elements.

If the request uses a technique called "URL Rewriting" to maintain sessions, then see section 6.1 Handling User Sessions Wit additional configuration steps.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this controller that is shown in the tree.
Server	Domain name or IP address of the web server. e.g. www.example.com. [Do not include the http:// prefix. in JMeter 2.5 (and later) if the "Host" header is defined in a Header Manager, then this will be used as the host name.
Port	Port the web server is listening to. Default: 80
Connect Timeout	Connection Timeout. Number of milliseconds to wait for a connection to open.

Response Timeout	Response Timeout. Number of milliseconds to wait for a response.
Server (proxy)	Hostname or IP address of a proxy server to perform request. [Do not include the http:// prefix.]
Port	Port the proxy server is listening to.
Username	(Optional) username for proxy server.
Password	(Optional) password for proxy server.
Implementation	Java, HttpClient3.1, HttpClient4. If not specified (and not defined by HTTP Request Defaults), the defaul depends on the value of the JMeter property `jmeter.httpsampler` , failing that, the Java implementatio used.
Protocol	HTTP, HTTPS or FILE. Default: HTTP
Method	GET, POST, HEAD, TRACE, OPTIONS, PUT, DELETE
Content Encoding	Content encoding to be used (for POST and FILE)
Redirect Automatically	Sets the underlying http protocol handler to automatically follow redirects, so they are not seen by JMete thus will not appear as samples. Should only be used for GET and HEAD requests. The HttpClient sampl reject attempts to use it for POST or PUT. **Warning: see below for information on cookie and header handling.**
Follow Redirects	This only has any effect if "Redirect Automatically" is not enabled. If set, the JMeter sampler will check response is a redirect and follow it if so. The initial redirect and further responses will appear as additiona samples. The URL and data fields of the parent sample will be taken from the final (non-redirected) samp the parent byte count and elapsed time include all samples. The latency is taken from the initial response (versions of JMeter after 2.3.4 - previously it was zero). Note that the HttpClient sampler may log the fol message: "Redirect requested but followRedirects is disabled" This can be ignored. In versions after 2.3.4, JMeter will collapse paths of the form '/../segment' in both absolute and relative re URLs. For example http://host/one/../two => http://host/two. If necessary, this behaviour can be suppress setting the JMeter property `httpsampler.redirect.removeslashdotdot=false`
Use KeepAlive	JMeter sets the Connection: keep-alive header. This does not work properly with the default HTTP implementation, as connection re-use is not under user-control. It does work with the Jakarta httpClient implementation.
Use multipart/form-data for HTTP POST	Use a multipart/form-data or application/x-www-form-urlencoded post request
Browser-compatible headers	When using multipart/form-data, this suppresses the Content-Type and Content-Transfer-Encoding heade the Content-Disposition header is sent.
Path	The path to resource (for example, /servlets/myServlet). If the resource requires query string parameters, them below in the "Send Parameters With the Request" section. **As a special case, if the path starts witl "http://" or "https://" then this is used as the full URL.** In this case, the server, port and protocol are i parameters are also ignored for GET and DELETE methods.
Send Parameters With the Request	The query string will be generated from the list of parameters you provide. Each parameter has a *name* ar *value* , the options to encode the parameter, and an option to include or exclude an equals sign (some applications don't expect an equals when the value is the empty string). The query string will be generate correct fashion, depending on the choice of "Method" you made (ie if you chose GET or DELETE, the qu string will be appended to the URL, if POST or PUT, then it will be sent separately). Also, if you are sen file using a multipart form, the query string will be created using the multipart form specifications. **See b for some further information on parameter handling.** Additionally, you can specify whether each parameter should be URL encoded. If you are not sure what t means, it is probably best to select it. If your values contain characters such as & or spaces, or question m then encoding is usually required.
File Path:	Name of the file to send. If left blank, JMeter does not send a file, if filled in, JMeter automatically sends request as a multipart form request. If it is a POST or PUT request and there is a single file whose 'name' attribute (below) is omitted, then th sent as the entire body of the request, i.e. no wrappers are added. This allows arbitrary bodies to be sent. functionality is present for POST requests after version 2.2, and also for PUT requests after version 2.3. **S below for some further information on parameter handling.**
Parameter name:	Value of the "name" web request parameter.
MIME Type	MIME type (for example, text/plain). If it is a POST or PUT request and either the 'name' atribute (below omitted or the request body is constructed from parameter values only, then the value of this field is used value of the content-type request header.
Retrieve All Embedded Resources from HTML Files	Tell JMeter to parse the HTML file and send HTTP/HTTPS requests for all images, Java applets, JavaSc files, CSSs, etc. referenced in the file. See below for more details.
Use as monitor	For use with the Monitor Results listener.
Save response as MD5 hash?	If this is selected, then the response is not stored in the sample result. Instead, the 32 character MD5 hash data is calculated and stored instead. This is intended for testing large amounts of data.

Embedded URLs must match:	If present, this must be a regular expression that is used to match against any embedded URLs found. So only want to download embedded resources from http://example.com/, use the expression: http://example\.com/.*
Use concurrent pool	Use a pool of concurrent connections to get embedded resources.
Size	Pool size for concurrent connections used to get embedded resources.
Source IP address:	[Only for HTTP Request HTTPClient] Override the default local IP address for this sample. The JMeter l must have multiple IP addresses (i.e. IP aliases or network interfaces). If the property **httpclient.localadd** defined, that is used for all HttpClient requests.

N.B. when using Automatic Redirection, cookies are only sent for the initial URL. This can cause unexpected behaviour for w to a local server. E.g. if www.example.com redirects to www.example.co.uk. In this case the server will probably return cooki JMeter will only see the cookies for the last host, i.e. www.example.co.uk. If the next request in the test plan uses www.examp www.example.co.uk, it will not get the correct cookies. Likewise, Headers are sent for the initial request, and won't be sent fo generally only a problem for manually created test plans, as a test plan created using a recorder would continue from the redir

Parameter Handling:

For the POST and PUT method, if there is no file to send, and the name(s) of the parameter(s) are omitted, then the body is cr all the value(s) of the parameters. This allows arbitrary bodies to be sent. The values are encoded if the encoding flag is set (v 2.3). See also the MIME Type above how you can control the content-type request header that is sent.

For other methods, if the name of the parameter is missing, then the parameter is ignored. This allows the use of optional para variables. (versions of JMeter after 2.3)

Method Handling:

The POST and PUT request methods work similarly, except that the PUT method does not support multipart requests. The PU be provided as one of the following:

- define the body as a file
- define the body as parameter value(s) with no name

If you define any parameters with a name in either the sampler or Http defaults then nothing is sent. The GET and DELETE r similarly to each other.

Upto and including JMeter 2.1.1, only responses with the content-type "text/html" were scanned for embedded resources. Oth assumed to be something other than HTML. JMeter 2.1.2 introduces the a new property **HTTPResponse.parsers** , which is a **htmlParser** and **wmlParser** . For each id found, JMeter checks two further properties:

- id.types - a list of content types
- id.className - the parser to be used to extract the embedded resources

See jmeter.properties file for the details of the settings. If the HTTPResponse.parser property is not set, JMeter reverts to the p only text/html responses will be scanned

Emulating slow connections (HttpClient only):

```
# Define characters per second > 0 to emulate slow connections
#httpclient.socket.http.cps=0
#httpclient.socket.https.cps=0
```

Response size calculation

Optional properties to allow change the method to get response size:

- Gets the real network size in bytes for the body response

  `sampleresult.getbytes.body_real_size=true`

- Add HTTP headers to full response size

  `sampleresult.getbytes.headers_size=true`

Versions of JMeter before 2.5 returns only data response size (uncompressed if request uses gzip/defate mode). To return to settings before version 2.5, set the two properties to false.

See Also:

- Assertion
- Building a Web Test Plan
- Building an Advanced Web Test Plan
- HTTP Authorization Manager
- HTTP Cookie Manager
- HTTP Header Manager
- HTML Link Parser
- HTTP Proxy Server
- HTTP Request Defaults
- HTTP Requests and Session ID's: URL Rewriting

18.1.3 JDBC Request

This sampler lets you send an JDBC Request (an SQL query) to a database.

Before using this you need to set up a JDBC Connection Configuration Configuration element

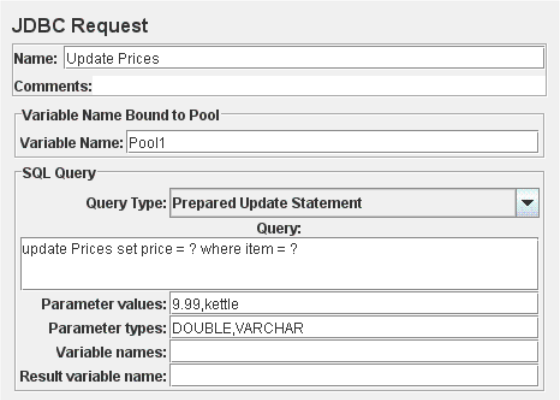
If the Variable Names list is provided, then for each row returned by a Select statement, the variables are set up with the value column (if a variable name is provided), and the count of rows is also set up. For example, if the Select statement returns 2 rov the variable list is `A,,C` , then the following variables will be set up:

```
A_#=2 (number of rows)
A_1=column 1, row 1
A_2=column 1, row 2
C_#=2 (number of rows)
C_1=column 3, row 1
C_2=column 3, row 2
```

If the Select statement returns zero rows, then the A_# and C_# variables would be set to 0, and no other variables would be s

Old variables are cleared if necessary - e.g. if the first select retrieves 6 rows and a second select returns only 3 rows, the addit rows 4, 5 and 6 will be removed.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this controller that is shown in the tree.
Variable Name	Name of the JMeter variable that the connection pool is bound to. This must agree with the 'Variable Name' field JDBC Connection Configuration.
Query Type	Set this according to the statement type: • Select Statement • Update Statement - use this for Inserts as well • Callable Statement • Prepared Select Statement • Prepared Update Statement - use this for Inserts as well • Commit • Rollback • Autocommit(false) • Autocommit(true) • Edit - this should be a variable reference that evaluates to one of the above
SQL Query	SQL query. Do not enter a trailing semi-colon. There is generally no need to use { and } to enclose Callable statements; however they mey be used if the database uses a non-standard syntax. [The JDBC driver automatical converts the statement if necessary when it is enclosed in {}]. For example: • select * from t_customers where id=23 • CALL SYSCS_UTIL.SYSCS_EXPORT_TABLE (null,?, ?, null, null, null) ◦ Parameter values: tablename,filename ◦ Parameter types: VARCHAR,VARCHAR The second example assumes you are using Apache Derby.
Parameter values	Comma-separated list of parameter values. Use]NULL[to indicate a NULL parameter. (If required, the null stri can be changed by defining the property "jdbcsampler.nullmarker".) The list must be enclosed in double-quotes if any of the values contain a comma or double-quote, and any embed double-quotes must be doubled-up, for example: `"Dbl-Quote: "" and Comma: ,"` There must be as many values as there are placeholders in the statement.

Parameter types	Comma-separated list of SQL parameter types (e.g. INTEGER, DATE, VARCHAR, DOUBLE). These are defin as fields in the class java.sql.Types, see for example: Javadoc for java.sql.Types . [Note: JMeter will use whateve types are defined by the runtime JVM, so if you are running on a different JVM, be sure to check the appropriate document] If the callable statement has INOUT or OUT parameters, then these must be indicated by prefixing th appropriate parameter types, e.g. instead of "INTEGER", use "INOUT INTEGER". If not specified, "IN" is assu i.e. "DATE" is the same as "IN DATE". If the type is not one of the fields found in java.sql.Types, versions of JMeter after 2.3.2 also accept the corresponding integer number, e.g. since INTEGER == 4, you can use "INOUT 4". There must be as many types as there are placeholders in the statement.
Variable Names	Comma-separated list of variable names to hold values returned by Select statements
Result Variable Name	If specified, this will create an Object variable containing a list of row maps. Each map contains the column nam the key and the column data as the value. Usage: `columnValue = vars.getObject("resultObject").get(0).get("Column Name");`

See Also:

- Building a Database Test Plan
- JDBC Connection Configuration

Versions of JMeter after 2.3.2 use UTF-8 as the character encoding. Previously the platform default was used.

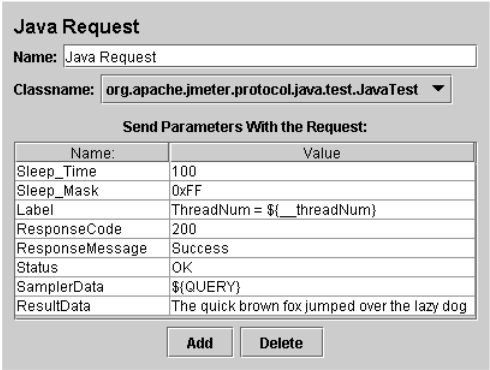
18.1.4 Java Request

This sampler lets you control a java class that implements the **`org.apache.jmeter.protocol.java.sampler.JavaSampler`** writing your own implementation of this interface, you can use JMeter to harness multiple threads, input parameter control, ar

The pull-down menu provides the list of all such implementations found by JMeter in its classpath. The parameters can then b below - as defined by your implementation. Two simple examples (JavaTest and SleepTest) are provided.

The JavaTest example sampler can be useful for checking test plans, because it allows one to set values in almost all the fields used by Assertions, etc. The fields allow variables to be used, so the values of these can readily be seen.

Control Panel



The Add/Delete buttons don't serve any purpose at present.

Parameters

Attribute	Description
Name	Descriptive name for this sampler that is shown in the tree.
Classname	The specific implementation of the JavaSamplerClient interface to be sampled.
Send Parameters with Request	A list of arguments that will be passed to the sampled class. All arguments are sent as Strings.

The sleep time is calculated as follows:

```
SleepTime is in milliseconds
SleepMask is used to add a "random" element to the time:
totalSleepTime = SleepTime + (System.currentTimeMillis() % SleepMask)
```

18.1.5 SOAP/XML-RPC Request

This sampler lets you send a SOAP request to a webservice. It can also be used to send XML-RPC over HTTP. It creates an H
with the specified XML as the POST content. To change the "Content-type" from the default of "text/xml", use a HeaderMana
sampler will use all the headers from the HeaderManager. If a SOAP action is specified, that will override any SOAPaction in
The primary difference between the soap sampler and webservice sampler, is the soap sampler uses raw post and does not req
SOAP 1.1.

For versions of JMeter later than 2.2, the sampler no longer uses chunked encoding by default.

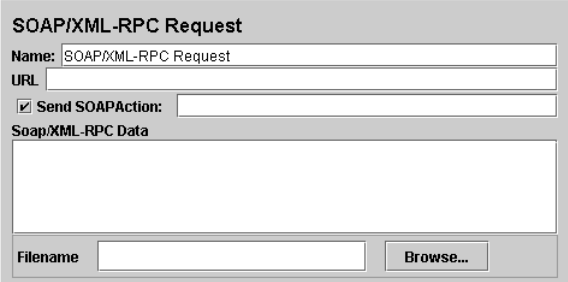
For screen input, it now always uses the size of the data.

File input uses the file length as determined by Java.

On some OSes this may not work for all files, in which case add a child Header Manager with Content-Length set to the actual length of the file.

Or set Content-Length to -1 to force chunked encoding.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this sampler that is shown in the tree.	No
URL	The URL to direct the SOAP request to.	Yes
Send SOAP action	Send a SOAP action header? (overrides the Header Manager)	No
Soap/XML-RPC Data	The Soap XML message, or XML-RPC instructions. Not used if the filename is provided.	No
Filename	If specified, then the contents of the file are sent, and the Data field is ignored	No

---

18.1.6 WebService(SOAP) Request

This sampler has been tested with IIS Webservice running .NET 1.0 and .NET 1.1. It has been tested with SUN JWSDP, IBM
gSoap toolkit for C/C++. The sampler uses Apache SOAP driver to serialize the message and set the header with the correct S
now the sampler doesn't support automatic WSDL handling, since Apache SOAP currently does not provide support for it. Bc
provide WSDL drivers. There are 3 options for the post data: text area, external file, or directory. If you want the sampler to ra
message, use the directory. Otherwise, use the text area or a file. The if either the file or path are set, it will not use the messag
you need to test a soap service that uses different encoding, use the file or path. If you paste the message in to text area, it will
encoding and will result in errors. Save your message to a file with the proper encoding, and the sampler will read it as java.io

An important note on the sampler is it will automatically use the proxy host and port passed to JMeter from command line, if
sampler are left blank. If a sampler has values in the proxy host and port text field, it will use the ones provided by the user. T
be what users expect.

By default, the webservice sampler sets SOAPHTTPConnection.setMaintainSession (true). If you need to maintain the sessio
Manager. The sampler uses the Header Manager to store the SOAPHTTPConnection object, since the version of apache soap
easy way to get and set the cookies.

Note: If you are using CSVDataSet, do not check "Memory Cache". If memory cache is checked, it will not iterate to the next
the requests will use the first value.

Make sure you use <soap:Envelope rather than <Envelope. For example:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<foo xmlns="http://clients-xlmns"/>
</soap:Body>
```

```
</soap:Envelope>
```

The SOAP library that is used does not support SOAP 1.2, only SOAP 1.1. Also the library does not provide access to the HTTP response code (e.g. 200) or message (e.g. OK). To get round this, versions of JMeter after 2.3.2 check the returned message length. If this is zero, then the request is marked as failed.

Control Panel

Parameters

Attribute	Description
Name	Descriptive name for this sampler that is shown in the tree.
WSDL URL	The WSDL URL with the service description. Versions of JMeter after 2.3.1 support the file: protocol for WSDL files.
Web Methods	Will be populated from the WSDL when the Load WSDL button is pressed. Select one of the methods anc Configure button to populate the Protocol, Server, Port, Path and SOAPAction fields.
Protocol	HTTP or HTTPS are acceptable protocol.
Server Name or IP	The hostname or IP address.
Port Number	Port Number.
Path	Path for the webservice.
SOAPAction	The SOAPAction defined in the webservice description or WSDL.
Soap/XML-RPC Data	The Soap XML message
Soap file	File containing soap message
Message Folder	Folder containing soap files
Memory cache	When using external files, setting this causes the file to be processed once and caches the result. This may memory if there are many different large files.
Use HTTP Proxy	Check box if http proxy should be used
Proxy Host	Proxy hostname
Proxy Port	Proxy host port

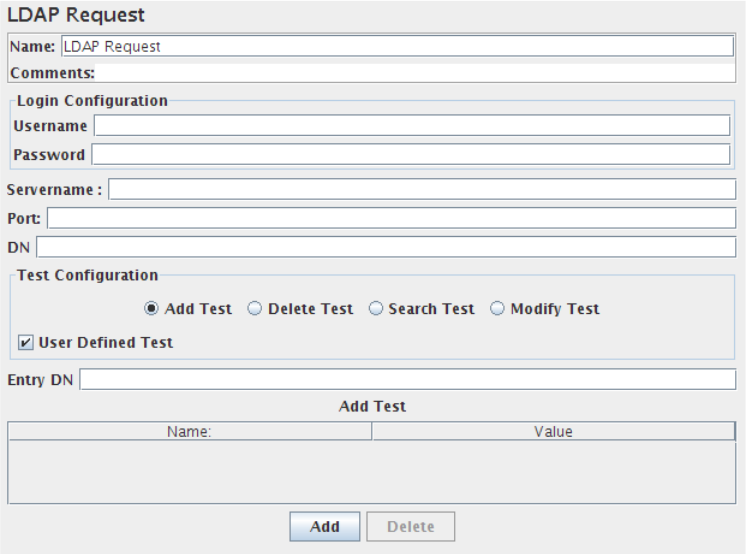
18.1.7 LDAP Request

This Sampler lets you send a different Ldap request(Add, Modify, Delete and Search) to an LDAP server.

If you are going to send multiple requests to the same LDAP server, consider using an LDAP Request Defaults Configuration have to enter the same information for each LDAP Request.

The same way the Login Config Element also using for Login and password.

Control Panel



There are two ways to create test cases for testing an LDAP Server.

1. Inbuilt Test cases.
2. User defined Test cases.

There are four test scenarios of testing LDAP. The tests are given below:

1. Add Test
    1. Inbuilt test :

       This will add a pre-defined entry in the LDAP Server and calculate the execution time. After execution of the test be deleted from the LDAP Server.

    2. User defined test :

       This will add the entry in the LDAP Server. User has to enter all the attributes in the table.The entries are collecte add. The execution time is calculated. The created entry will not be deleted after the test.

2. Modify Test
    1. Inbuilt test :

       This will create a pre-defined entry first, then will modify the created entry in the LDAP Server.And calculate the execution of the test, the created entry will be deleted from the LDAP Server.

    2. User defined test

       This will modify the entry in the LDAP Server. User has to enter all the attributes in the table. The entries are col modify. The execution time is calculated. The entry will not be deleted from the LDAP Server.

3. Search Test
    1. Inbuilt test :

       This will create the entry first, then will search if the attributes are available. It calculates the execution time of th end of the execution,created entry will be deleted from the LDAP Server.

    2. User defined test

       This will search the user defined entry(Search filter) in the Search base (again, defined by the user). The entries sl the LDAP Server. The execution time is calculated.

4. Delete Test
    1. Inbuilt test :

       This will create a pre-defined entry first, then it will be deleted from the LDAP Server. The execution time is calc

2. User defined test

   This will delete the user-defined entry in the LDAP Server. The entries should be available in the LDAP Server. T
   calculated.

Parameters

Attribute	Description
Name	Descriptive name for this controller that is shown in the tree.
Server Name or IP	Domain name or IP address of the LDAP server. JMeter assumes the LDAP server is listening on the defaul
Port	default port(389).
root DN	DN for the server to communicate
Username	LDAP server username.
Password	LDAP server password.
Entry DN	the name of the context to create or Modify; may not be empty Example: do you want to add cn=apache,ou= have to add in table name=cn, value=apache
Delete	the name of the context to Delete; may not be empty
Search base	the name of the context or object to search
Search filter	the filter expression to use for the search; may not be null
add test	this name, value pair to added in the given context object
modify test	this name, value pair to add or modify in the given context object

See Also:

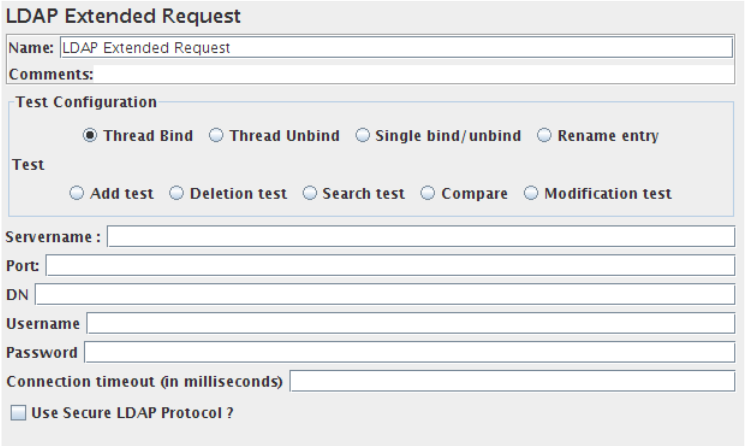
- Building an Ldap Test Plan
- LDAP Request Defaults

18.1.8 LDAP Extended Request

This Sampler can send all 8 different LDAP request to an LDAP server. It is an extended version of the LDAP sampler, theref configure, but can be made much closer resembling a real LDAP session.

If you are going to send multiple requests to the same LDAP server, consider using an LDAP Extended Request Defaults Con you do not have to enter the same information for each LDAP Request.

Control Panel



There are nine test operations defined. These operations are given below:

1. **Thread bind**

   Any LDAP request is part of an LDAP session, so the first thing that should be done is starting a session to the LDAP s session a thread bind is used, which is equal to the LDAP "bind" operation. The user is requested to give a username (D and password, which will be used to initiate a session. When no password, or the wrong password is specified, an anony started. Take care, omitting the password will not fail this test, a wrong password will.

   Parameters

Attribute	Description
Name	Descriptive name for this sampler that is shown in the tree.
Servername	The name (or IP-address) of the LDAP server.
Port	The port number that the LDAP server is listening to. If this is omitted JMeter assumes the LDAP server i on the default port(389).

DN	The distinguished name of the base object that will be used for any subsequent operation. It can be used as point for all operations. You cannot start any operation on a higher level than this DN!
Username	Full distinguished name of the user as which you want to bind.
Password	Password for the above user. If omitted it will result in an anonymous bind. If is is incorrect, the sampler v an error and revert to an anonymous bind.

2. **Thread unbind**

   This is simply the operation to end a session. It is equal to the LDAP "unbind" operation.

   Parameters

Attribute	Description	Required
Name	Descriptive name for this sampler that is shown in the tree.	No

3. **Single bind/unbind**

   This is a combination of the LDAP "bind" and "unbind" operations. It can be used for an authentication request/passwo It will open an new session, just to check the validity of the user/password combination, and end the session again.

   Parameters

Attribute	Description
Name	Descriptive name for this sampler that is shown in the tree.
Username	Full distinguished name of the user as which you want to bind.
Password	Password for the above user. If omitted it will result in an anonymous bind. If is is incorrect, the sampler wi error.

4. **Rename entry**

   This is the LDAP "moddn" operation. It can be used to rename an entry, but also for moving an entry or a complete sub in the LDAP tree.

   Parameters

Attribute	Description
Name	Descriptive name for this sampler that is shown in the tree.
Old entry name	The current distinguished name of the object you want to rename or move, relative to the given thread bind operation.
New distinguished name	The new distinguished name of the object you want to rename or move, relative to the given DN thread bind operation.

5. **Add test**

   This is the ldap "add" operation. It can be used to add any kind of object to the LDAP server.

   Parameters

Attribute	Description
Name	Descriptive name for this sampler that is shown in the tree.
Entry DN	Distinguished name of the object you want to add, relative to the given DN in the thread bind operation.
Add test	A list of attributes and their values you want to use for the object. If you need to add a multiple value attribu need to add the same attribute with their respective values several times to the list.

6. **Delete test**

   This is the LDAP "delete" operation, it can be used to delete an object from the LDAP tree

   Parameters

Attribute	Description	R
Name	Descriptive name for this sampler that is shown in the tree.	N
Delete	Distinguished name of the object you want to delete, relative to the given DN in the thread bind operation.	Y

7. **Search test**

   This is the LDAP "search" operation, and will be used for defining searches.

Parameters

Attribute	Description
Name	Descriptive name for this sampler that is shown in the tree.
Search base	Distinguished name of the subtree you want your search to look in, relative to the given DN in the thre operation.
Search Filter	searchfilter, must be specified in LDAP syntax.
Scope	Use 0 for baseobject-, 1 for onelevel- and 2 for a subtree search. (Default=0)
Size Limit	Specify the maximum number of results you want back from the server. (default=0, which means no l When the sampler hits the maximum number of results, it will fail with errorcode 4
Time Limit	Specify the maximum amount of (cpu)time (in miliseconds) that the server can spend on your search. this does not say anything about the responsetime. (default is 0, which means no limit)
Attributes	Specify the attributes you want to have returned, seperated by a semicolon. An empty field will return attributes
Return object	Whether the object will be returned (true) or not (false). Default=false
Dereference aliases	If true, it will dereference aliases, if false, it will not follow them (default=false)

8. **Modification test**

   This is the LDAP "modify" operation. It can be used to modify an object. It can be used to add, delete or replace values

   Parameters

Attribute	Description
Name	Descriptive name for this sampler that is shown in the tree.
Entry name	Distinguished name of the object you want to modify, relative to the given DN in the thread bind operat
Modification test	The attribute-value-opCode triples. The opCode can be any valid LDAP operationCode (add, delete/ren replace). If you don't specify a value with a delete operation, all values of the given attribute will be del do specify a value in a delete operation, only the given value will be deleted. If this value is non-existen sampler will fail the test.

9. **Compare**

   This is the LDAP "compare" operation. It can be used to compare the value of a given attribute with some already know is mostly used to check whether a given person is a member of some group. In such a case you can compare the DN of t value, with the values in the attribute "member" of an object of the type groupOfNames. If the compare operation fails, errorcode 49.

   Parameters

Attribute	Description
Name	Descriptive name for this sampler that is shown in the tree.
Entry DN	The current distinguished name of the object of which you want to compare an attribute, relative to the g the thread bind operation.
Compare filter	In the form "attribute=value"

See Also:

- Building an LDAP Test Plan
- LDAP Extended Request Defaults

18.1.9 Access Log Sampler

(Alpha Code)

AccessLogSampler was designed to read access logs and generate http requests. For those not familiar with the access log, it i maintains of every request it accepted. This means the every image and html file. The current implementation is complete, but not been enabled. There is a filter for the access log parser, but I haven't figured out how to link to the pre-processor. Once I d sampler will be made to enable that functionality.

Tomcat uses the common format for access logs. This means any webserver that uses the common log format can use the Acc Server that use common log format include: Tomcat, Resin, Weblogic, and SunOne. Common log format looks like this:

127.0.0.1 - - [21/Oct/2003:05:37:21 -0500] "GET /index.jsp?%2Findex.jsp= HTTP/1.1" 200 8343

The current implemenation of the parser only looks at the text within the quotes. Everything else is stripped out and igored. Fo
response code is completely ignored by the parser. For the future, it might be nice to filter out entries that do not have a respor
Extending the sampler should be fairly simple. There are two interfaces you have to implement.

org.apache.jmeter.protocol.http.util.accesslog.LogParser

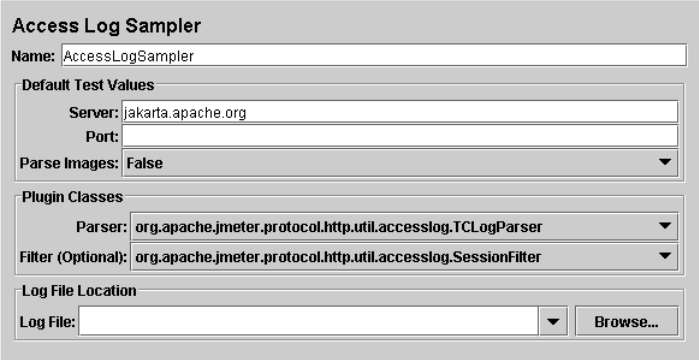
org.apache.jmeter.protocol.http.util.accesslog.Generator

The current implementation of AccessLogSampler uses the generator to create a new HTTPSampler. The servername, port an
AccessLogSampler. Next, the parser is called with integer 1, telling it to parse one entry. After that, HTTPSampler.sample() is
request.

```
samp = (HTTPSampler) GENERATOR.generateRequest();
samp.setDomain(this.getDomain());
samp.setPort(this.getPort());
samp.setImageParser(this.isImageParser());
PARSER.parse(1);
res = samp.sample();
res.setSampleLabel(samp.toString());
```

The required methods in LogParser are: setGenerator(Generator) and parse(int). Classes implementing Generator interface sho
implementation for all the methods. For an example of how to implement either interface, refer to StandardGenerator and TCI

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this controller that is shown in the tree.	No
Server	Domain name or IP address of the web server.	Yes
Port	Port the web server is listening to.	No (defaults to 80)
Log parser class	The log parser class is responsible for parsing the logs.	Yes (default provided)
Filter	The filter class is used to filter out certain lines.	No
Location of log file	The location of the access log file.	Yes

The TCLogParser processes the access log independently for each thread. The SharedTCLogParser and OrderPreservingLogP
the file, i.e. each thread gets the next entry in the log.

The SessionFilter is intended to handle Cookies across threads. It does not filter out any entries, but modifies the cookie mana
for a given IP are processed by a single thread at a time. If two threads try to process samples from the same client IP address,
forced to wait until the other has completed.

The LogFilter is intended to allow access log entries to be filtered by filename and regex, as well as allowing for the replacem
However, it is not currently possible to configure this via the GUI, so it cannot really be used.

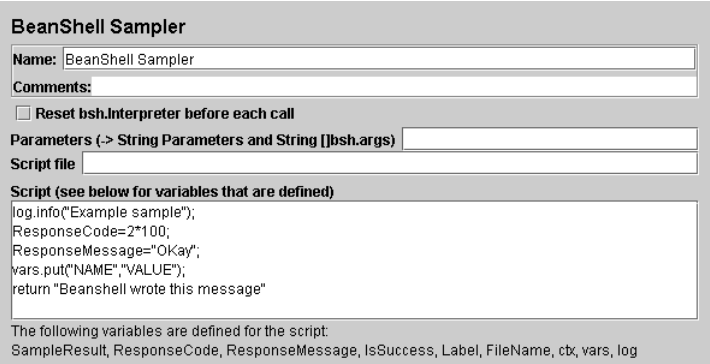
18.1.10 BeanShell Sampler

This sampler allows you to write a sampler using the BeanShell scripting language.

For full details on using BeanShell, please see the BeanShell website.

The test element supports the ThreadListener and TestListener methods. These should be defined in the initialisation file. See
BeanShellListeners.bshrc for example definitions.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this controller that is shown in the tree. The name is stored in the script variable Label
Reset bsh.Interpreter before each call	If this option is selected, then the interpreter will be recreated for each sample. This may be necessary for some long running scripts. For further information, see Best Practices - BeanShell scripting .
Parameters	Parameters to pass to the BeanShell script. This is intended for use with script files; for scripts define in the GUI, you can use whatever variable and function references you need within the script itself. T parameters are stored in the following variables: • Parameters - string containing the parameters as a single variable • bsh.args - String array containing parameters, split on white-space
Script file	A file containing the BeanShell script to run. The file name is stored in the script variable FileName
Script	The BeanShell script to run. The return value (if not null) is stored as the sampler result.

N.B. Each Sampler instance has its own BeanShell interpeter, and Samplers are only called from a single thread

If the property "beanshell.sampler.init" is defined, it is passed to the Interpreter as the name of a sourced file. This can be used methods and variables. There is a sample init file in the bin directory: BeanShellSampler.bshrc.

If a script file is supplied, that will be used, otherwise the script will be used.

Before invoking the script, some variables are set up in the BeanShell interpreter:

The contents of the Parameters field is put into the variable "Parameters". The string is also split into separate tokens using a s separator, and the resulting list is stored in the String array bsh.args.

The full list of BeanShell variables that is set up is as follows:

- log - the Logger
- Label - the Sampler label
- FileName - the file name, if any
- Parameters - text from the Parameters field
- bsh.args - the parameters, split as described above
- SampleResult - pointer to the current SampleResult
- ResponseCode = 200
- ResponseMessage = "OK"
- IsSuccess = true
- ctx - JMeterContext
- vars - JMeterVariables - e.g. vars.get("VAR1"); vars.put("VAR2","value"); vars.remove("VAR3"); vars.putObject("OB
- props - JMeterProperties - e.g. props.get("START.HMS"); props.put("PROP1","1234");

When the script completes, control is returned to the Sampler, and it copies the contents of the following script variables into t variables in the SampleResult:

- ResponseCode - for example 200
- ResponseMessage - for example "OK"
- IsSuccess - true/false

The SampleResult ResponseData is set from the return value of the script. Since version 2.1.2, if the script returns null, it can directly, by using the method SampleResult.setResponseData(data), where data is either a String or a byte array. The data type can be set to binary by using the method SampleResult.setDataType(SampleResult.BINARY).

The SampleResult variable gives the script full access to all the fields and methods in the SampleResult. For example, the scri methods setStopThread(boolean) and setStopTest(boolean). Here is a simple (not very useful!) example script:

```
if (bsh.args[0].equalsIgnoreCase("StopThread")) {
    log.info("Stop Thread detected!");
    SampleResult.setStopThread(true);
}
return "Data from sample with Label "+Label;
//or, since version 2.1.2
SampleResult.setResponseData("My data");
return null;
```

Another example:

ensure that the property **beanshell.sampler.init=BeanShellSampler.bshrc** is defined in jmeter.properties. The following scri
of all the variables in the ResponseData field:

```
return getVariables();
```

For details on the methods available for the various classes (JMeterVariables, SampleResult etc) please check the Javadoc or t
Beware however that misuse of any methods can cause subtle faults that may be difficult to find ...

18.1.11 BSF Sampler

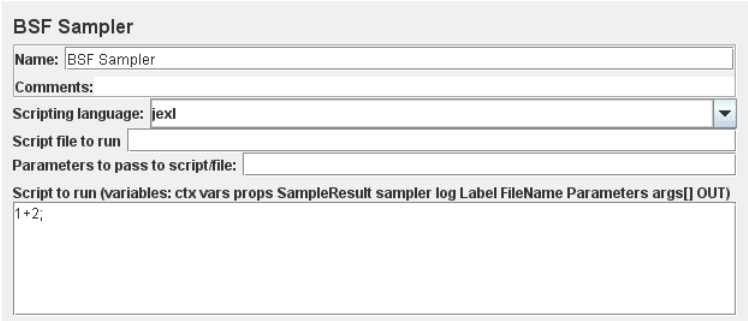
This sampler allows you to write a sampler using a BSF scripting language.

See the Apache Bean Scripting Framework website for details of the languages supported. You may need to download the app
language; they should be put in the JMeter **lib** directory.

By default, JMeter supports the following languages:

- javascript
- jexl (JMeter version 2.3.2 and later)
- xslt

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this controller that is shown in the tree.
Scripting Language	Name of the BSF scripting language to be used. N.B. Not all the languages in the drop-down list are supporte by default. The following are supported: jexl, javascript, xslt. Others may be available if the appropriate jar is installed in the JMeter lib directory.
Script File	Name of a file to be used as a BSF script
Parameters	List of parameters to be passed to the script file or the script.
Script	Script to be passed to BSF language

If a script file is supplied, that will be used, otherwise the script will be used.

Before invoking the script, some variables are set up. Note that these are BSF variables - i.e. they can be used directly in the s

- log - the Logger
- Label - the Sampler label
- FileName - the file name, if any
- Parameters - text from the Parameters field
- args - the parameters, split as described above
- SampleResult - pointer to the current SampleResult
- sampler - pointer to current Sampler
- ctx - JMeterContext
- vars - JMeterVariables - e.g. vars.get("VAR1"); vars.put("VAR2","value"); vars.remove("VAR3"); vars.putObject("OB
- props - JMeterProperties - e.g. props.get("START.HMS"); props.put("PROP1","1234");
- OUT - System.out - e.g. OUT.println("message")

The SampleResult ResponseData is set from the return value of the script. If the script returns null, it can set the response dire
method SampleResult.setResponseData(data), where data is either a String or a byte array. The data type defaults to "text", bu
by using the method SampleResult.setDataType(SampleResult.BINARY).

The SampleResult variable gives the script full access to all the fields and methods in the SampleResult. For example, the scri
methods setStopThread(boolean) and setStopTest(boolean).

Unlike the Beanshell Sampler, the BSF Sampler does not set the ResponseCode, ResponseMessage and sample status via scri
the only way to changes these is via the SampleResult methods:

- SampleResult.setSuccessful(true/false)
- SampleResult.setResponseCode("code")
- SampleResult.setResponseMessage("message")

18.1.11.1 JSR223 Sampler

The JSR223 Sampler allows JSR223 script code to be used to perform a sample. For details, see BSF Sampler .

18.1.12 TCP Sampler

The TCP Sampler opens a TCP/IP connection to the specified server. It then sends the text, and waits for a response.

If "Re-use connection" is selected, connections are shared between Samplers in the same thread, provided that the exact same port are used. Different hosts/port combinations will use different connections, as will different threads.

If an error is detected - or "Re-use connection" is not selected - the socket is closed. Another socket will be reopened on the ne

The following properties can be used to control its operation:

- tcp.status.prefix - text that precedes a status number
- tcp.status.suffix - text that follows a status number
- tcp.status.properties - name of property file to convert status codes to messages
- tcp.handler - Name of TCP Handler class (default TCPClientImpl) - only used if not specified on the GUI

The class that handles the connection is defined by the GUI, failing that the property tcp.handler. If not found, the class is ther package org.apache.jmeter.protocol.tcp.sampler.

Users can provide their own implementation. The class must extend org.apache.jmeter.protocol.tcp.sampler.TCPClient.

The following implementations are currently provided.

- TCPClientImpl
- BinaryTCPClientImpl
- LengthPrefixedBinaryTCPClientImpl

The implementations behave as follows:

TCPClientImpl

This implementation is fairly basic. When reading the response, it reads until the end of line byte, if this is defined by setting t **tcp.eolByte** , otherwise until the end of the input stream.

BinaryTCPClientImpl

This implementation converts the GUI input, which must be a hex-encoded string, into binary, and performs the reverse when When reading the response, it reads until the end of message byte, if this is defined by setting the property **tcp.BinaryTCPCli** otherwise until the end of the input stream.

LengthPrefixedBinaryTCPClientImpl

This implementation extends BinaryTCPClientImpl by prefixing the binary message data with a binary length byte. The lengt bytes. This can be changed by setting the property **tcp.binarylength.prefix.length** .

Timeout handling If the timeout is set, the read will be terminated when this expires. So if you are using an eolByte/eomByte timeout is sufficiently long, otherwise the read will be terminated early.

Response handling

If tcp.status.prefix is defined, then the response message is searched for the text following that up to the suffix. If any such tex set the response code. The response message is then fetched from the properties file (if provided).

For example, if the prefix = "[" and the suffix = "]", then the following repsonse:

[J28] XI123,23,GBP,CR

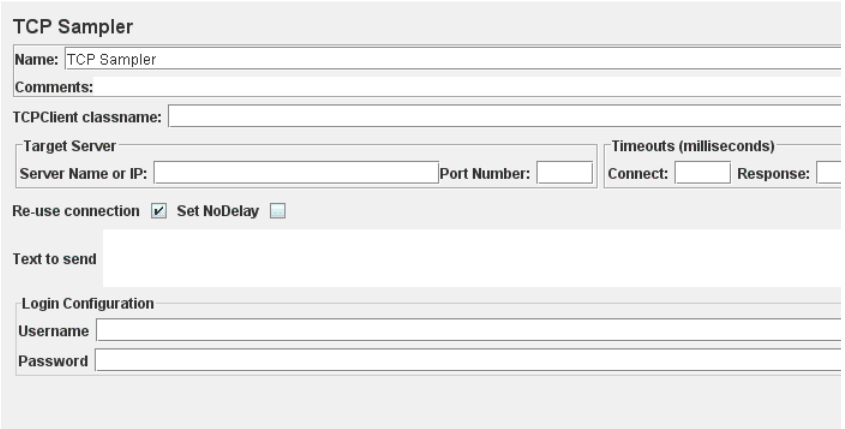
would have the response code J28.

Response codes in the range "400"-"499" and "500"-"599" are currently regarded as failures; all others are successful. [This n configurable!]

The login name/password are not used by the supplied TCP implementations.

Sockets are disconnected at the end of a test run.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
TCPClient classname	Name of the TCPClient class. Defaults to the property tcp.handler, failing that TCPClientImpl.
ServerName or IP	Name or IP of TCP server
Port Number	Port to be used
Re-use connection	If selected, the connection is kept open. Otherwise it is closed when the data has been read.
Connect Timeout	Connect Timeout (milliseconds, 0 disables).
Response Timeout	Response Timeout (milliseconds, 0 disables).
Set Nodelay	See java.net.Socket.setTcpNoDelay(). If selected, this will disable Nagle's algorithm, otherwise Nagle's will be used.
Text to Send	Text to be sent
Login User	User Name - not used by default implementation
Password	Password - not used by default implementation

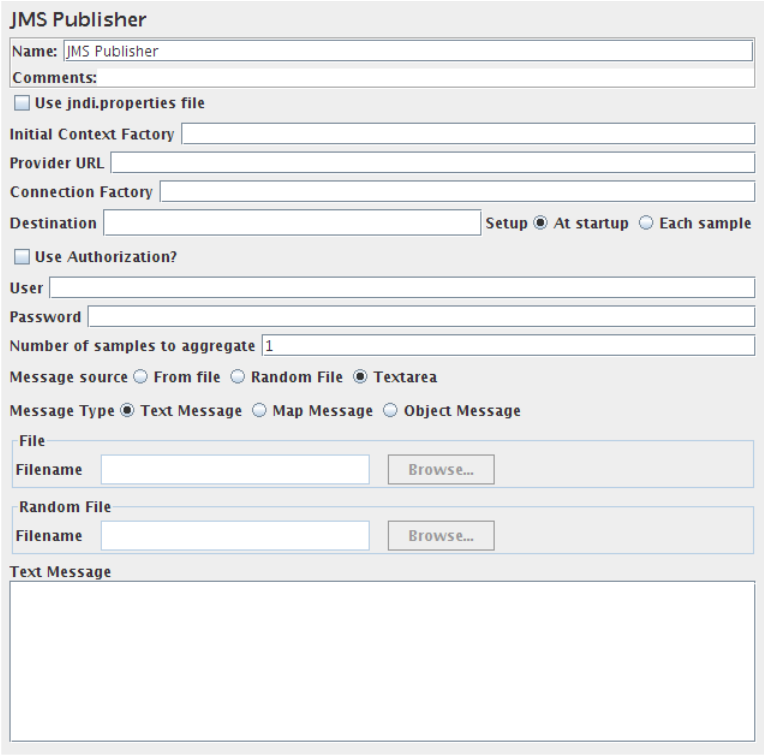
18.1.13 JMS Publisher

BETA CODE - the code is still subject to change

JMS Publisher will publish messages to a given destination (topic/queue). For those not familiar with JMS, it is the J2EE spec messaging. There are numerous JMS servers on the market and several open source options.

JMeter does not include any JMS implementation jar; this must be downloaded from the JMS provider and put in the lib directory

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
use JNDI properties file	use jndi.properties. Note that the file must be on the classpath - e.g. by updating the user.classpath JMeter property. If this option is not selected, JMeter uses the "JNDI Initial Context Factory" and "Provider URL" fields to create the connection.
JNDI Initial Context Factory	Name of the context factory
Provider URL	The URL for the jms provider
Destination	The message destination (topic or queue name)
Setup	The destination setup type. With At startup, the destination name is static (i.e. always same name during the test), with Each sample, the destination name is dynamic and is evaluate at each sample (i.e. the destination name may be a variable)
Authentication	Authentication requirement for the JMS provider
User	User Name
Password	Password
Number of samples to aggregate	Number of samples to aggregate
Message source	Where to obtain the message
Message type	Text, Map or Object message

For the MapMessage type, JMeter reads the source as lines of text. Each line must have 3 fields, delimited by commas. The fi

- Name of entry
- Object class name, e.g. "String" (assumes java.lang package if not specified)
- Object string value

For each entry, JMeter adds an Object with the given name. The value is derived by creating an instance of the class, and usin method to convert the value if necessary. For example:

```
name,String,Example
size,Integer,1234
```

This is a very simple implementation; it is not intended to support all possible object types.

Note: the Object message type is not implemented yet.

The following table shows some values which may be useful when configuring JMS:

Apache ActiveMQ	Value(s)	Comment
Context Factory	org.apache.activemq.jndi.ActiveMQInitialContextFactory	.
Provider URL	vm://localhost	

Provider URL	vm:(broker:(vm://localhost)?persistent=false)	Disable persistence
Queue Reference	dynamicQueues/QUEUENAME	Dynamically define the QUEUENAME to JNDI
Topic Reference	dynamicTopics/TOPICNAME	Dynamically define the TOPICNAME to JNDI

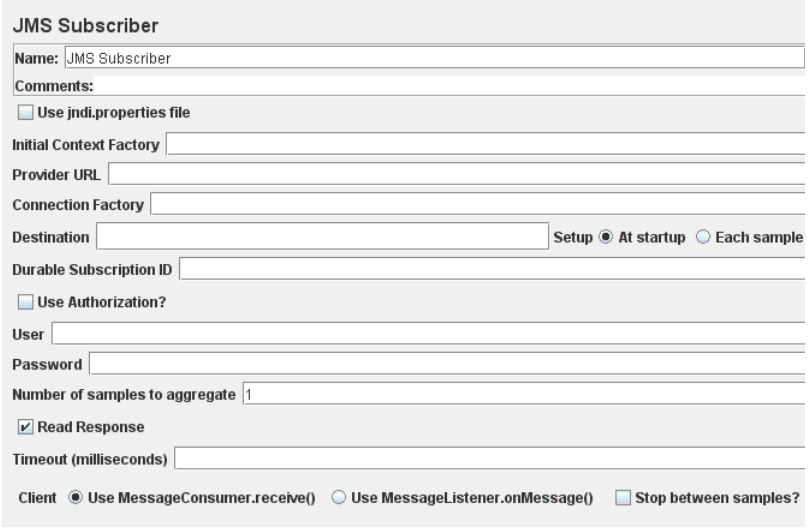
18.1.14 JMS Subscriber

BETA CODE - the code is still subject to change

JMS Publisher will subscribe to messages in a given destination (topic or queue). For those not familiar with JMS, it is the J2I messaging. There are numerous JMS servers on the market and several open source options.

JMeter does not include any JMS implementation jar; this must be downloaded from the JMS provider and put in the lib directory

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
use JNDI properties file	use jndi.properties. Note that the file must be on the classpath - e.g. by updating the user.classpath JMete If this option is not selected, JMeter uses the "JNDI Initial Context Factory" and "Provider URL" fields t connection.
JNDI Initial Context Factory	Name of the context factory
Provider URL	The URL for the jms provider
Destination	the message destination (topic or queue name)
Durable Subscription ID	The ID to use for a durable subscription. On first use the respective queue will automatically be generate JMS provider if it does not exist yet.
Setup	The destination setup type. With At startup, the destination name is static (i.e. always same name during with Each sample, the destination name is dynamic and is evaluate at each sample (i.e. the destination na a variable)
Authentication	Authentication requirement for the JMS provider
User	User Name
Password	Password
Number of samples to aggregate	number of samples to aggregate
Read response	should the sampler read the response. If not, only the response length is returned.
Timeout	Specify the timeout to be applied, in milliseconds. 0=none. This is the overall aggregate timeout, not per
Client	Which client implementation to use. Both of them create connections which can read messages. Howeve different strategy, as described below: • MessageConsumer.receive() - calls receive() for every requested message. Retains the connection samples, but does not fetch messages unless the sampler is active. This is best suited to Queue sub

	• MessageListener.onMessage() - establishes a Listener that stores all incoming messages on a queu listener remains active after the sampler completes. This is best suited to Topic subscriptions.
Stop between samples?	If selected, then JMeter calls Connection.stop() at the end of each sample (and calls start() before each sa This may be useful in some cases where multiple samples/threads have connections to the same queue. I selected, JMeter calls Connection.start() at the start of the thread, and does not call stop() until the end of

NOTE: JMeter 2.3.4 and earlier used a different strategy for the MessageConsumer.receive() client. Previously this started a l which polled for messages. This thread continued when the sampler completed, so the net effect was similar to the MessageLi strategy.

18.1.15 JMS Point-to-Point

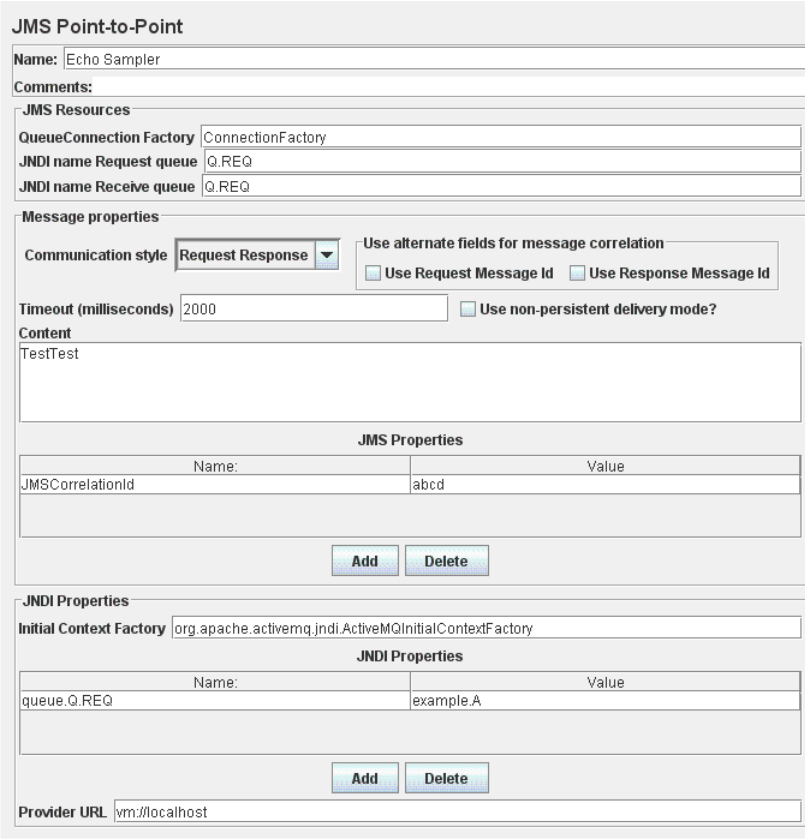
BETA CODE - the code is still subject to change

This sampler sends and optionally receives JMS Messages through point-to-point connections (queues). It is different from pu generally used for handling transactions.

Versions of JMeter after 2.3.2 use the properties java.naming.security.[principal|credentials] - if present - when creating the Q this behaviour is not desired, set the JMeter property **JMSSampler.useSecurity.properties=false**

JMeter does not include any JMS implementation jar; this must be downloaded from the JMS provider and put in the lib directory

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
QueueConnection Factory	The JNDI name of the queue connection factory to use for connecting to the messaging system.
JNDI Name Request queue	This is the JNDI name of the queue to which the messages are sent.

JNDI Name Reply queue	The JNDI name of the receiving queue. If a value is provided here and the communication style is Requ Response this queue will be monitored for responses to the requests sent.
Communication style	The Communication style can be Request Only (also known as Fire and Forget) or Request Reply. Requ will only sent messages and will not monitor replies. As such it can be used to put load on a system. Re will sent messages and monitor the replies it receives. Behaviour is depended on the value of the JNDI I Reply Queue. If JNDI Name Reply Queue has a value, this queue is used to monitor the results. Matchi request and reply is done with the message id of the request with the correlation id of the reply. If the JN Reply Queue is empty, then temporary queues will be used for the communication between the requestc server. This is very different from the fixed reply queue. With temporary queues the diffent threads will the reply message has been received.
Use alternate fields for message correlation	These check-boxes select the fields which will be used for matching the response message with the orig request. • Use Request Message Id - if selected, the request JMSMessageID will be used, otherwise the requ JMSCorrelationID will be used. In the latter case the correlation id must be specified in the reque: • Use Response Message Id - if selected, the response JMSMessageID will be used, otherwise the r JMSCorrelationID will be used. There are two frequently used JMS Correlation patterns: • JMS Correlation ID Pattern - i.e. match request and response on their correlation Ids => deselect l checkboxes, and provide a correlation id. • JMS Message ID Pattern - i.e. match request message id with response correlation id => select "U Message Id" only. In both cases the JMS application is responsible for populating the correlation ID as necessary. **Note:** if queue is used to send and receive messages, then the response message will be the same as the request r which case, either provide a correlation id and clear both checkboxes; or select both checkboxes to use t message Id for correlation. This can be useful for checking raw JMS throughput.
Timeout	The timeout in milliseconds for the reply-messages. If a reply has not been received within the specified specific testcase failes and the specific reply message received after the timeout is discarded.
Use non-persistent delivery mode?	Whether to set DeliveryMode.NON_PERSISTENT.
Content	The content of the message.
JMS Properties	The JMS Properties are properties specific for the underlying messaging system. For example: for WebS web services you will need to set the JMS Property targetService to test webservices through JMS.
Initial Context Factory	The Initial Context Factory is the factory to be used to look up the JMS Resources.
JNDI properties	The JNDI Properties are the specific properties for the underlying JNDI implementation.
Provider URL	The URL for the jms provider.

18.1.16 JUnit Request

The current implementation supports standard Junit convention and extensions. It also includes extensions like oneTimeSetUp oneTimeTearDown. The sampler works like the JavaSampler with some differences.

1. rather than use Jmeter's test interface, it scans the jar files for classes extending junit's TestCase class. That includes any cla

2. Junit test jar files should be placed in jmeter/lib/junit instead of /lib directory. In versions of JMeter after 2.3.1, you can alsc "user.classpath" property to specify where to look for TestCase classes.

3. Junit sampler does not use name/value pairs for configuration like the JavaSampler. The sampler assumes setUp and tearDc test correctly.

4. The sampler measures the elapsed time only for the test method and does not include setUp and tearDown.

5. Each time the test method is called, Jmeter will pass the result to the listeners.

6. Support for oneTimeSetUp and oneTimeTearDown is done as a method. Since Jmeter is multi-threaded, we cannot call oneTimeSetUp/oneTimeTearDown the same way Maven does it.

7. The sampler reports unexpected exceptions as errors. There are some important differences between standard JUnit test run implementation. Rather than make a new instance of the class for each test, JMeter creates 1 instance per sampler and reuses i

The current implementation of the sampler will try to create an instance using the string constructor first. If the test class does constructor, the sampler will look for an empty constructor. Example below:

Empty Constructor:
```
public class myTestCase {
public myTestCase() {}
}
```

String Constructor:
```
public class myTestCase {
public myTestCase(String text) {
super(text);
}
}
```

By default, Jmeter will provide some default values for the success/failure code and message. Users should define a set of uni
codes and use them uniformly across all tests.
General Guidelines

If you use setUp and tearDown, make sure the methods are declared public. If you do not, the test may not run properly.

Here are some general guidelines for writing Junit tests so they work well with Jmeter. Since Jmeter runs multi-threaded, it is
certain things in mind.

1. Write the setUp and tearDown methods so they are thread safe. This generally means avoid using static memebers.
2. Make the test methods discrete units of work and not long sequences of actions. By keeping the test method to a descrete o
easier to combine test methods to create new test plans.
3. Avoid making test methods depend on each other. Since Jmeter allows arbitrary sequencing of test methods, the runtime be
the default Junit behavior.
4. If a test method is configurable, be careful about where the properties are stored. Reading the properties from the Jar file is
5. Each sampler creates an instance of the test class, so write your test so the setup happens in oneTimeSetUp and oneTimeTe

Control Panel

Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Search for JUnit4 annotations	Select this to search for JUnit 4 tests (@Test annotations)
Package filter	Comma separated list of packages to show. Example, org.apache.jmeter,junit.framework.
Class name	Fully qualified name of the JUnit test class.
Constructor string	String pass to the string constructor. If a string is set, the sampler will use the string constructor instead of t constructor.
Test method	The method to test.
Success message	A descriptive message indicating what success means.
Success code	An unique code indicating the test was successful.
Failure message	A descriptive message indicating what failure means.
Failure code	An unique code indicating the test failed.
Error message	A description for errors.
Error code	Some code for errors. Does not need to be unique.
Do not call setUp and tearDown	Set the sampler not to call setUp and tearDown. By default, setUp and tearDown should be called. Not call methods could affect the test and make it inaccurate. This option should only be used with calling oneTime oneTimeTearDown. If the selected method is oneTimeSetUp or oneTimeTearDown, this option should be
Append assertion errors	Whether or not to append assertion errors to the response message.
Append runtime exceptions	Whether or not to append runtime exceptions to the response message. Only applies if "Append assertion e selected.

The following JUnit4 annotations are recognised:

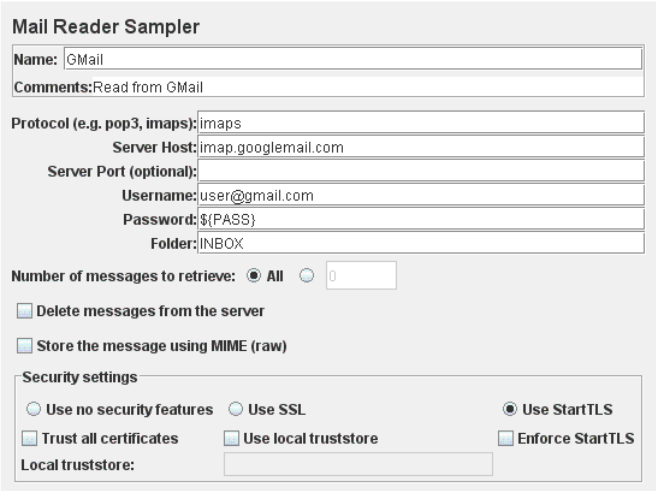
- @Test - used to find test methods and classes. The "expected" and "timeout" attributes are supported.
- @Before - treated the same as setUp() in JUnit3
- @After - treated the same as tearDown() in JUnit3
- @BeforeClass, @AfterClass - treated as test methods so they can be run independently as required

Note that JMeter currently runs the test methods directly, rather than leaving it to JUnit. This is to allow the setUp/tearDown r
from the sample time.

18.1.17 Mail Reader Sampler

The Mail Reader Sampler can read (and optionally delete) mail messages using POP3(S) or IMAP(S) protocols.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Server Type	The protocol used by the provider: e.g. pop3, pop3s, imap, imaps. or another string representing the serve protocol. For example `file` for use with the read-only mail file provider. The actual provider names for and IMAP are `pop3` and `imap`
Server	Hostname or IP address of the server. See below for use with `file` protocol.
Port	Port to be used to connect to the server (optional)
Username	User login name
Password	User login password (N.B. this is stored unencrypted in the test plan)
Folder	The IMAP(S) folder to use. See below for use with `file` protocol.
Number of messages to retrieve	Set this to retrieve all or some messages
Delete messages from the server	If set, messages will be deleted after retrieval
Store the message using MIME	Whether to store the message as MIME. If so, then the entire raw message is stored in the Response Data headers are not stored as they are available in the data. If not, the message headers are stored as Response Headers. A few headers are stored (Date, To, From, Subject) in the body.
Use no security features	Indicates that the connection to the server does not use any security protocol.
Use SSL	Indicates that the connection to the server must use the SSL protocol.
Use StartTLS	Indicates that the connection to the server should attempt to start the TLS protocol.
Enforce StartTLS	If the server does not start the TLS protocol the connection will be terminated.
Trust All Certificates	When selected it will accept all certificates independent of the CA.
Use local truststore	When selected it will only accept certificates that are locally trusted.
Local truststore	Path to file containing the trusted certificates. Relative paths are resolved against the current directory. Failing that, against the directory containing the test script (JMX file).

Messages are stored as subsamples of the main sampler. In versions of JMeter after 2.3.4, multipart message parts are stored a
message.

Special handling for "file" protocol:

The `file` JavaMail provider can be used to read raw messages from files. The `server` field is used to specify the path to the

Individual message files should be stored with the name `n.msg` , where `n` is the message number. Alternatively, the `server f` of a file which contains a single message. The current implementation is quite basic, and is mainly intended for debugging pu

---

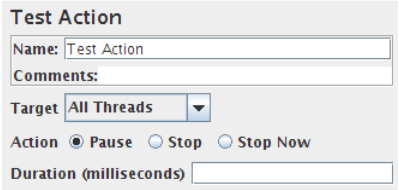
18.1.18 Test Action

The Test Action sampler is a sampler that is intended for use in a conditional controller. Rather than generate a sample, the te pauses or stops the selected target.

This sampler can also be useful in conjunction with the Transaction Controller, as it allows pauses to be included without nee sample. For variable delays, set the pause time to zero, and add a Timer as a child.

The "Stop" action stops the thread or test after completing any samples that are in progress. The "Stop Now" action stops the t samples to complete; it will interrupt any active samples. If some threads fail to stop within the 5 second time-limit, a messag GUI mode. You can try using the Stop command to see if this will stop the threads, but if not, you should exit JMeter. In non- will exit if some threads fail to stop within the 5 second time limit. [This can be changed using the JMeter property `jmeterengine.threadstop.wait`]

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this element that is shown in the tree.	No
Target	Current Thread / All Threads (ignored for Pause)	Yes
Action	Pause / Stop / Stop Now	Yes
Duration	How long to pause for (milliseconds)	Yes, if Pause is selected

---

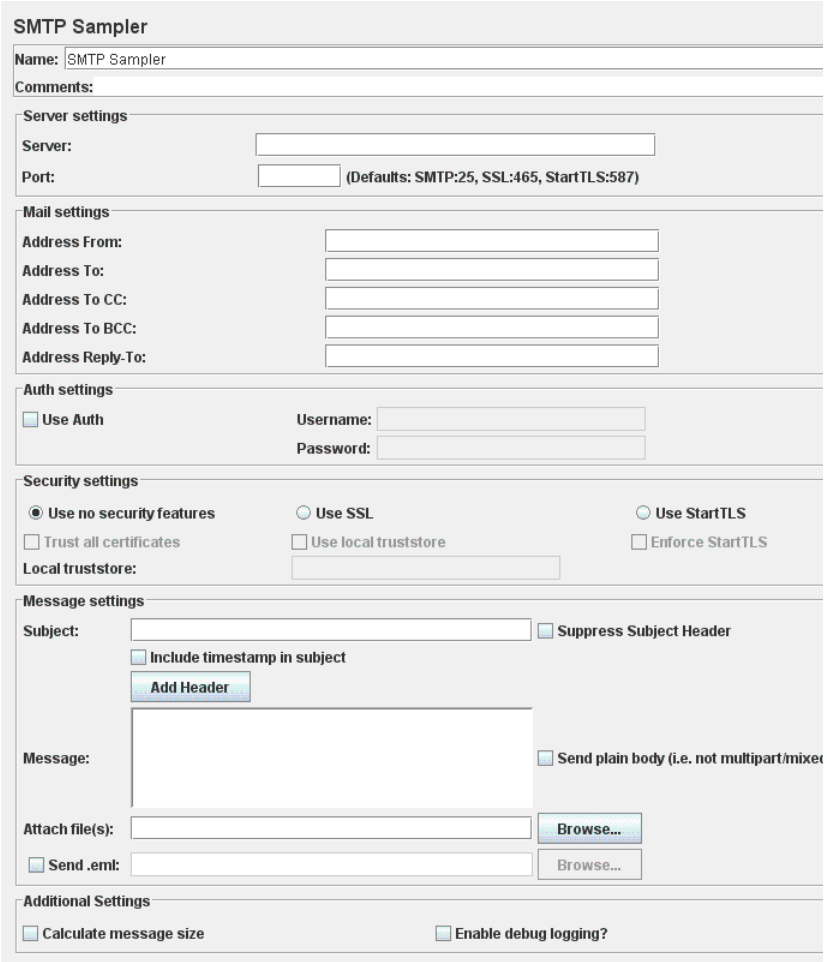
18.1.19 SMTP Sampler

The SMTP Sampler can send mail messages using SMTP/SMTPS protocol. It is possible to set security propocols for the con TLS), as well as user authentication. If a security protocol is used a verification on the server certificate will occur.

Two alternatives to handle this verification are available:

- Trust all certificates. This will ignore certificate chain verification
- Use a local truststore. With this option the certificate chain will be validated against the local truststore file.

Control Panel



Parameters

Attribute	Description
Server	Hostname or IP address of the server. See below for use with `file` protocol.
Port	Port to be used to connect to the server. Defaults are: SMTP=25, SSL=465, StartTLS=587
Address From	The from address that will appear in the e-mail
Address To	The destination e-mail address (multiple values separated by ";")
Address To CC	Carbon copy destinations e-mail address (multiple values separated by ";")
Address To BCC	Blind carbon copy destinations e-mail address (multiple values separated by ";")
Address Reply-To	Alternate Reply-To address (multiple values separated by ";")
Use Auth	Indicates if the SMTP server requires user authentication
Username	User login name
Password	User login password (N.B. this is stored unencrypted in the test plan)
Use no security features	Indicates that the connection to the SMTP server does not use any security protocol.
Use SSL	Indicates that the connection to the SMTP server must use the SSL protocol.
Use StartTLS	Indicates that the connection to the SMTP server should attempt to start the TLS protocol.
Enforce StartTLS	If the server does not start the TLS protocol the connection will be terminated.
Trust All Certificates	When selected it will accept all certificates independent of the CA.
Use local truststore	When selected it will only accept certificates that are locally trusted.
Local truststore	Path to file containing the trusted certificates. Relative paths are resolved against the current director Failing that, against the directory containing the test script (JMX file).
Subject	The e-mail message subject.
Suppress Subject Header	If selected, the "Subject:" header is omitted from the mail that is sent. This is different from sending empty "Subject:" header, though some e-mail clients may display it identically.
Include timestamp in subject	Includes the System.currentTimemilis() in the subject line.
Add Header	Additional headers can be defined using this button.

Message	The message body.
Send plain body (i.e. not multipart/mixed)	If selected, then send the body as a plain message, i.e. not multipart/mixed, if possible. If the messag body is empty and there is a single file, then send the file contents as the message body. Note: If the message body is not empty, and there is at least one attached file, then the body is sent as multipart/mixed.
Attach files	Files to be attached to the message.
Send .eml	If set, the .eml file will be sent instead of the entries in the Subject, Message, and Attached files
Calculate message size	Calculates the message size and stores it in the sample result.
Enable debug logging?	If set, then the "mail.debug" property is set to "true"

---

^

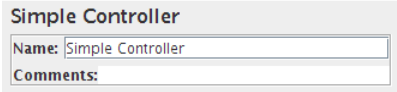
18.2 Logic Controllers

Logic Controllers determine the order in which Samplers are processed.

18.2.1 Simple Controller

The Simple Logic Controller lets you organize your Samplers and other Logic Controllers. Unlike other Logic Controllers, thi no functionality beyond that of a storage device.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this controller that is shown in the tree.	No

Using the Simple Controller

Download this example (see Figure 6). In this example, we created a Test Plan that sends two Ant HTTP requests and two Lo We grouped the Ant and Log4J requests by placing them inside Simple Logic Controllers. Remember, the Simple Logic Cont how JMeter processes the controller(s) you add to it. So, in this example, JMeter sends the requests in the following order: An News Page, Log4J Home Page, Log4J History Page. Note, the File Reporter is configured to store the results in a file named " current directory.

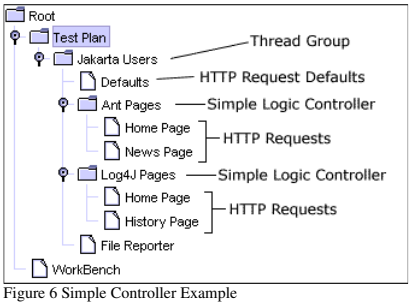


Figure 6 Simple Controller Example

18.2.2 Loop Controller

If you add Generative or Logic Controllers to a Loop Controller, JMeter will loop through them a certain number of times, in value you specified for the Thread Group. For example, if you add one HTTP Request to a Loop Controller with a loop count the Thread Group loop count to three, JMeter will send a total of 2 * 3 = 6 HTTP Requests.

Control Panel

Parameters

Attribute	Description
Name	Descriptive name for this controller that is shown in the tree.
Loop Count	The number of times the subelements of this controller will be iterated each time through a test run. **Special Case:** The Loop Controller embedded in the Thread Group element behaves slightly differently. Unless set to forever, it stops the test after the given number of iterations have been done.

Looping Example

Download this example (see Figure 4). In this example, we created a Test Plan that sends a particular HTTP Request only onc HTTP Request five times.

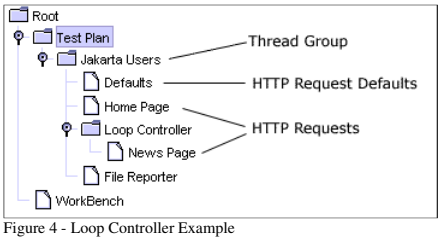


Figure 4 - Loop Controller Example

We configured the Thread Group for a single thread and a loop count value of one. Instead of letting the Thread Group contro a Loop Controller. You can see that we added one HTTP Request to the Thread Group and another HTTP Request to a Loop (configured the Loop Controller with a loop count value of five.

JMeter will send the requests in the following order: Home Page, News Page, News Page, News Page, News Page, and News Reporter is configured to store the results in a file named "loop-test.dat" in the current directory.

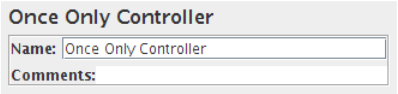
18.2.3 Once Only Controller

The Once Only Logic Controller tells JMeter to process the controller(s) inside it only once, and pass over any requests under iterations through the test plan.

The Once Only Controller will now execute always during the first iteration of any looping parent controller. Thus, if the Onc placed under a Loop Controller specified to loop 5 times, then the Once Only Controller will execute only on the first iteratior Controller (ie, every 5 times). Note this means the Once Only Controller will still behave as previously expected if put under a only once per test), but now the user has more flexibility in the use of the Once Only Controller.

For testing that requires a login, consider placing the login request in this controller since each thread only needs to login once

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this controller that is shown in the tree.	No

Once Only Example

Download this example (see Figure 5). In this example, we created a Test Plan that has two threads that send HTTP request. E request to the Home Page, followed by three requests to the Bug Page. Although we configured the Thread Group to iterate th thread only sends one request to the Home Page because this request lives inside a Once Only Controller.

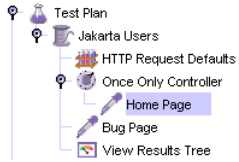


Figure 5. Once Only Controller Example

Each JMeter thread will send the requests in the following order: Home Page, Bug Page, Bug Page, Bug Page. Note, the File I to store the results in a file named "loop-test.dat" in the current directory.

The behaviour of the Once Only controller under anything other than the Thread Group or a Loop Controller is not currently defined. Odd things may happen.

18.2.4 Interleave Controller

If you add Generative or Logic Controllers to an Interleave Controller, JMeter will alternate among each of the other controlle iteration.

Control Panel

Parameters

Attribute	Description
name	Descriptive name for this controller that is shown in the tree.
ignore sub-controller blocks	If checked, the interleave controller will treat sub-controllers like single request elements and only a request per controller at a time.

Simple Interleave Example

Download this example (see Figure 1). In this example, we configured the Thread Group to have two threads and a loop count ten requests per thread. See the table below for the sequence JMeter sends the HTTP Requests.

Figure 1 - Interleave Controller Example 1

Loop Iteration	Each JMeter Thread Sends These HTTP Requests
1	News Page
1	Log Page
2	FAQ Page
2	Log Page
3	Gump Page
3	Log Page
4	Because there are no more requests in the controller,
	JMeter starts over and sends the first HTTP Request, which is the News Page.
4	Log Page
5	FAQ Page
5	Log Page

Useful Interleave Example

Download another example (see Figure 2). In this example, we configured the Thread Group to have a single thread and a loo Notice that the Test Plan has an outer Interleave Controller with two Interleave Controllers inside of it.

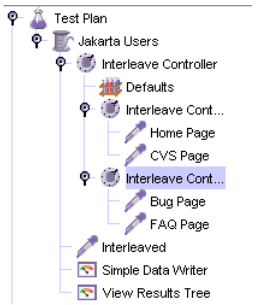


Figure 2 - Interleave Controller Example 2

The outer Interleave Controller alternates between the two inner ones. Then, each inner Interleave Controller alternates betwe Requests. Each JMeter thread will send the requests in the following order: Home Page, Interleaved, Bug Page, Interleaved, C and FAQ Page, Interleaved. Note, the File Reporter is configured to store the results in a file named "interleave-test2.dat" in tl

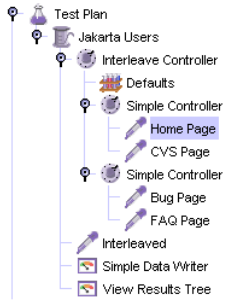


Figure 3 - Interleave Controller Example 3

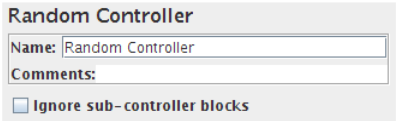
If the two interleave controllers under the main interleave controller were instead simple controllers, then the order would be: Page, Interleaved, Bug Page, FAQ Page, Interleaved. However, if "ignore sub-controller blocks" was checked on the main int the order would be: Home Page, Interleaved, Bug Page, Interleaved, CVS Page, Interleaved, and FAQ Page, Interleaved.

18.2.5 Random Controller

The Random Logic Controller acts similarly to the Interleave Controller, except that instead of going in order through its sub-samplers, it picks one at random at each pass.

Interactions between multiple controllers can yield complex behavior. This is particularly true of the Random Controller. Experiment before you assume what results any given interaction will give

Control Panel



Parameters

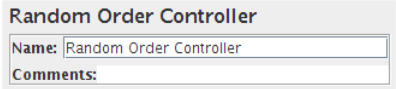
Attribute	Description	Required
Name	Descriptive name for this controller that is shown in the tree.	No

---

18.2.6 Random Order Controller

The Random Order Controller is much like a Simple Controller in that it will execute each child element at most once, but the the nodes will be random.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this controller that is shown in the tree.	No

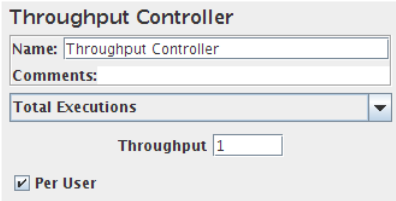
---

18.2.7 Throughput Controller

This controller is badly named, as it does not control throughput. Please refer to the Constant Throughput Timer for an el to adjust the throughput.

The Throughput Controller allows the user to control how often it is executed. There are two modes - percent execution and t executions causes the controller to execute a certain percentage of the iterations through the test plan. Total executions causes executing after a certain number of executions have occurred. Like the Once Only Controller, this setting is reset when a pare restarts.

Control Panel



The Throughput Controller can yield very complex behavior when combined with other controllers - in particular with interleave or random controllers as parents (also very useful).

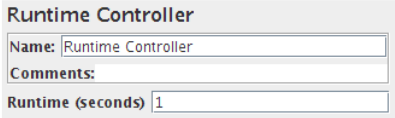
Parameters

Attribute	Description
Name	Descriptive name for this controller that is shown in the tree.
Execution Style	Whether the controller will run in percent executions or total executions mode.
Throughput	A number. for percent execution mode, a number from 0-100 that indicates the percentage of times the controll execute. "50" means the controller will execute during half the iterations throught the test plan. for total executi the number indicates the total number of times the controller will execute.
Per User	If checked, per user will cause the controller to calculate whether it should execute on a per user (per thread) ba unchecked, then the calculation will be global for all users. for example, if using total execution mode, and uncl user", then the number given for throughput will be the total number of executions made. if "per user" is checke total number of executions would be the number of users times the number given for throughput.

18.2.8 Runtime Controller

The Runtime Controller controls how long its children are allowed to run.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this controller that is shown in the tree, and used to name the transaction.	Yes
Runtime (seconds)	Desired runtime in seconds	Yes

18.2.9 If Controller

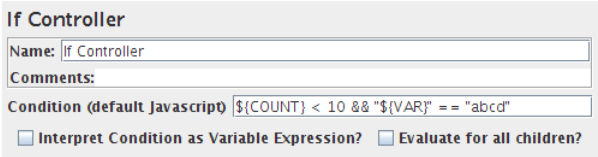
The If Controller allows the user to control whether the test elements below it (its children) are run or not.

Prior to JMeter 2.3RC3, the condition was evaluated for every runnable element contained in the controller. This sometimes c behaviour, so 2.3RC3 was changed to evaluate the condition only once on initial entry. However, the original behaviour is als JMeter after 2.3RC4 have an additional option to select the original behaviour.

Versions of JMeter after 2.3.2 allow the script to be processed as a variable expression, rather than requiring Javascript. It was use functions and variables in the Javascript condition, so long as they evaluated to "true" or "false"; now this can be done wit using Javascript as well. For example, previously one could use the condition: `${__jexl(${VAR} == 23)}` and this would be true/false, the result would then be passed to Javascript which would then return true/false. If the Variable Expression option i expression is evaluated and compared with "true", without needing to use Javascript. Also, variable expressions can return any Javascript condition must return "true"/"false" or an error is logged.

No variables are made available to the script when the condition is interpreted as Javascript. If you need access to such variables, then select "Interpret Condition as Variable Expression?" and use a __javaScript() function call. You can then use the objects "vars", "log", "ctx" etc. in the script.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this controller that is shown in the tree.
Condition (default Javascript)	By default the condition is interpreted as **Javascript** code that returns "true" or "false", but this can overriden (see below)
Interpret Condition as Variable Expression?	If this is selected, then the condition must be an expression that evaluates to "true" (case is ignored) example, `${FOUND}` or `${__jexl(${VAR} > 100)}` . Unlike the Javascript case, the condition is o checked to see if it matches "true" (case is ignored).
Evaluate for all children	Should condition be evaluated for all children? If not checked, then the condition is only evaluated

Examples (Javascript):

- ${COUNT} < 10
- "${VAR}" == "abcd"
- ${JMeterThread.last_sample_ok} (check if last sample succeeded)

If there is an error interpreting the code, the condition is assumed to be false, and a message is logged in jmeter.log.

Examples (Variable Expression):

- ${__jexl(${COUNT} < 10)}
- ${RESULT}

18.2.10 While Controller

The While Controller runs its children until the condition is "false".

Possible condition values:

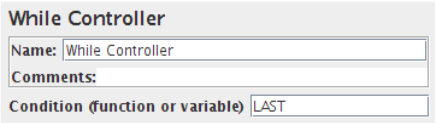
- blank - exit loop when last sample in loop fails
- LAST - exit loop when last sample in loop fails. If the last sample just before the loop failed, don't enter loop.
- Otherwise - exit (or don't enter) the loop when the condition is equal to the string "false"

The condition can be any variable or function that eventually evaluates to the string "false". This allows the use of JavaScript, BeanShell, properties or variables as needed.

For example:

- ${VAR} - where VAR is set to false by some other test element
- ${__javaScript(${C}==10)}
- ${__javaScript("${VAR2}"=="abcd")}
- ${_P(property)} - where property is set to "false" somewhere else

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this controller that is shown in the tree, and used to name the transaction.	Yes
Condition	blank, LAST, or variable/function	Yes

18.2.11 Switch Controller

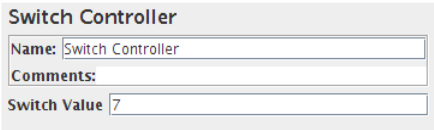
The Switch Controller acts like the Interleave Controller in that it runs one of the subordinate elements on each iteration, but r sequence, the controller runs the element defined by the switch value.

Note: In versions of JMeter after 2.3.1, the switch value can also be a name.

If the switch value is out of range, it will run the zeroth element, which therefore acts as the default for the numeric case. It als element if the value is the empty string.

If the value is non-numeric (and non-empty), then the Switch Controller looks for the element with the same name (case is sig the names match, then the element named "default" (case not significant) is selected. If there is no default, then no element is controller will not run anything.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this controller that is shown in the tree, and used to name the transaction.	Yes
Switch Value	The number (or name) of the subordinate element to be invoked. Elements are numbered from 0.	Yes

---

18.2.12 ForEach Controller

A ForEach controller loops through the values of a set of related variables. When you add samplers (or controllers) to a ForEa sample sample (or controller) is executed one or more times, where during every loop the variable has a new value. The input several variables, each extended with an underscore and a number. Each such variable must have a value. So for example whe has the name inputVar, the following variables should have been defined:

- inputVar_1 = wendy
- inputVar_2 = charles
- inputVar_3 = peter
- inputVar_4 = john

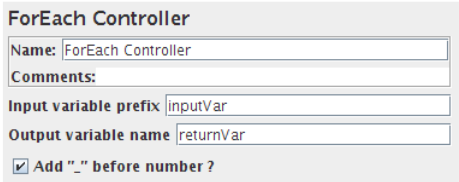
Note: the "_" separator is now optional.

When the return variable is given as "returnVar", the collection of samplers and controllers under the ForEach controller will consecutive times, with the return variable having the respective above values, which can then be used in the samplers.

It is especially suited for running with the regular expression post-processor. This can "create" the necessary input variables o a previous request. By omitting the "_" separator, the ForEach Controller can be used to loop through the groups by using the refName_g, and can also loop through all the groups in all the matches by using an input variable of the form refName_${C}_ counter variable.

The ForEach Controller does not run any samples if inputVar_1 is null. This would be the case if the Regular Expression returned no matches.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this controller that is shown in the tree.	No
Input variable prefix	Prefix for the variable names to be used as input.	Yes
Output variable	The name of the variable which can be used in the loop for replacement in the samplers	Yes
Use Separator	If not checked, the "_" separator is omitted.	Yes

ForEach Example

Download this example (see Figure 7). In this example, we created a Test Plan that sends a particular HTTP Request only onc
HTTP Request to every link that can be found on the page.

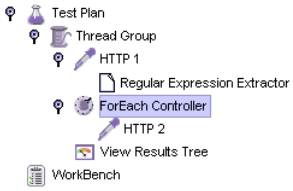


Figure 7 - ForEach Controller Example

We configured the Thread Group for a single thread and a loop count value of one. You can see that we added one HTTP Req
Group and another HTTP Request to the ForEach Controller.

After the first HTTP request, a regular expression extractor is added, which extracts all the html links out of the return page ar
inputVar variable

In the ForEach loop, a HTTP sampler is added which requests all the links that were extracted from the first returned HTML p

ForEach Example

Here is another example you can download. This has two Regular Expressions and ForEach Controllers. The first RE matches
not match, so no samples are run by the second ForEach Controller

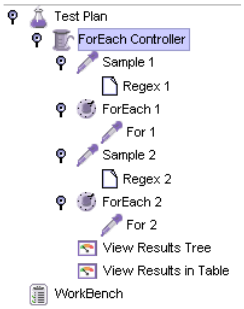


Figure 8 - ForEach Controller Example 2

The Thread Group has a single thread and a loop count of two.

Sample 1 uses the JavaTest Sampler to return the string "a b c d".

The Regex Extractor uses the expression **(\w)\s** which matches a letter followed by a space, and returns the letter (not the spac
prefixed with the string "inputVar".

The ForEach Controller extracts all variables with the prefix "inputVar_", and executes its sample, passing the value in the va
this case it will set the variable to the values "a" "b" and "c" in turn.

The For 1 Sampler is another Java Sampler which uses the return variable "returnVar" as part of the sample Label and as the s

Sample 2, Regex 2 and For 2 are almost identical, except that the Regex has been changed to "(\w)\sx", which clearly won't m
Sampler will not be run.

18.2.13 Module Controller

The Module Controller provides a mechanism for substituting test plan fragments into the current test plan at run-time.

A test plan fragment consists of a Controller and all the test elements (samplers etc) contained in it. The fragment can be locat
Group, or on the WorkBench . If the fragment is located in a Thread Group, then its Controller can be disabled to prevent the
except by the Module Controller. Or you can store the fragments in a dummy Thread Group, and disable the entire Thread Gr

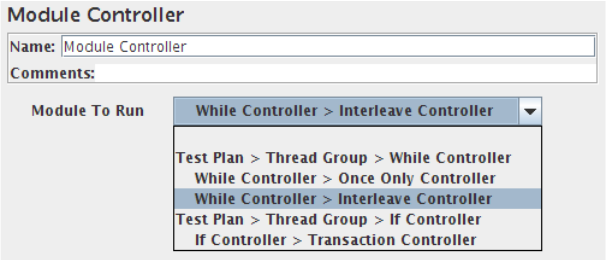
There can be multiple fragments, each with a different series of samplers under them. The module controller can then be used
between these multiple test cases simply by choosing the appropriate controller in its drop down box. This provides convenier
alternate test plans quickly and easily.

A fragment name is made up of the Controller name and all its parent names. For example:

```
Test Plan / Protocol: JDBC / Control / Interleave Controller (Module1)
```

Any **fragments used by the Module Controller must have a unique name** , as the name is used to find the target controller
reloaded. For this reason it is best to ensure that the Controller name is changed from the default - as shown in the example ab
duplicate may be accidentally created when new elements are added to the test plan.

Control Panel



The Module Controller should not be used with remote testing or non-gui testing in conjunction with Workbench components since the Workbench test elements are not part of test plan .jmx files. Any such test will fail.

Parameters

Attribute	Description
Name	Descriptive name for this controller that is shown in the tree.
Module to Run	The module controller provides a list of all controllers loaded into the gui. Select the one you want to substitu runtime.

18.2.14 Include Controller

The include controller is designed to use an external jmx file. To use it, create a Test Fragment underneath the Test Plan and a samplers, controllers etc. below it. Then save the Test Plan. The file is now ready to be included as part of other Test Plans.

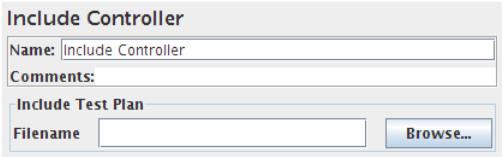
For convenience, a Thread Group can also be added in the external JMX file for debugging purposes. A Module Controller ca the Test Fragment. The Thread Group will be ignored during the include process.

If the test uses a Cookie Manager or User Defined Variables, these should be placed in the top-level test plan, not the includec are not guaranteed to work.

This element does not support variables/functions in the filename field. However, if the property **includecontroller.prefix** is defined, the contents are used to prefix the pathname.

If the file cannot be found at the location given by prefix+filename, then the controller attempts to open the fileName relative directory (versions of JMeter after 2.3.4).

Control Panel



Parameters

Attribute	Description	Required
Filename	The file to include.	Yes

18.2.15 Transaction Controller

The Transaction Controller generates an additional sample which measures the overall time taken to perform the nested test el time includes all processing within the controller scope, not just the samples.

For JMeter versions after 2.3, there are two modes of operation

- additional sample is added after the nested samples
- additional sample is added as a parent of the nested samples

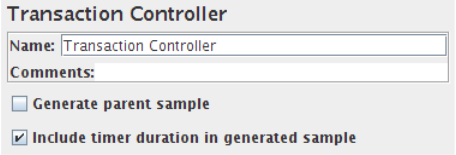
The generated sample time includes all the times for the nested samplers, **and any timers etc.** Depending on the clock resolut longer than the sum of the individual samplers plus timers. The clock might tick after the controller recorded the start time but sample starts. Similarly at the end.

The generated sample is only regarded as successful if all its sub-samples are successful.

In parent mode, the individual samples can still be seen in the Tree View Listener, but no longer appear as separate entries in
the sub-samples do not appear in CSV log files, but they can be saved to XML files.

In parent mode, Assertions (etc) can be added to the Transaction Controller. However by default they will be applied to both the individual samples and the overall transaction sample. To limit the scope of the Assertions, use a Simple Controller to contain the samples, and add the Assertions to the Simple Controller. Parent mode controllers do not currently properly support nested transaction controllers of either type.

Control Panel



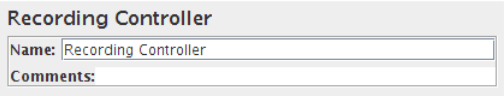
Parameters

Attribute	Description
Name	Descriptive name for this controller that is shown in the tree, and used to name the transaction
Generate Parent Sample	If checked, then the sample is generated as a parent of the other samples, otherwise the sample generated as an independent sample.
Include timer duration in generated sample	Whether to include timer, pre- and post-processing delays in the generated sample. Default is t compatible with the behaviour in previous versions of JMeter.

18.2.16 Recording Controller

The Recording Controller is a place holder indicating where the proxy server should record samples to. During test run, it has
the Simple Controller. But during recording using the HTTP Proxy Server , all recorded samples will by default be saved unde
Controller.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this controller that is shown in the tree.	No

---

^

18.3 Listeners

Most of the listeners perform several roles in addition to "listening" to the test results. They also provide means to view, save,
results.

Note that Listeners are processed at the end of the scope in which they are found.

The saving and reading of test results is generic. The various listeners have a panel whereby one can specify the file to which t
written (or read from). By default, the results are stored as XML files, typically with a ".jtl" extension. Storing as CSV is the n
but is less detailed than XML (the other available option).

Listeners do *not* process sample data in non-GUI mode, but the raw data will be saved if an output file has been config
analyse the data generated by a non-GUI test run, you need to load the file into the appropriate Listener.

To read existing results and display them, use the file panel Browse button to open the file.

Versions of JMeter up to 2.3.2 **used to clear any current data** before loading the new file.

This is no longer done, thus **allowing files to be merged** . If the previous behaviour is required, use the menu item Run/Clear
Run/Clear All (Ctrl+E) before loading the file.

Results can be read from XML or CSV format files. When reading from CSV results files, the header (if present) is used to de
are present. **In order to interpret a header-less CSV file correctly, the appropriate properties must be set in jmeter.prop**

The file name can contain function and/or variable references. However variable references do not work in client-server mode (functions work OK).

Listeners can use a lot of memory if there are a lot of samples. Most of the listeners currently keep a copy of every sample
from:

- Simple Data Writer
- BeanShell/BSF Listener
- Mailer Visualizer
- Monitor Results
- Summary Report

The following Listeners no longer need to keep copies of every single sample. Instead, samples with the same elapsed time are
memory is now needed, especially if most samples only take a second or two at most.

- Aggregate Report
- Aggregate Graph
- Distribution Graph

To minimise the amount of memory needed, use the Simple Data Writer, and use the CSV format.

Versions of JMeter after 2.3.1 allow JMeter variables to be saved to the output files. This can only be specified using a property. See the Listener Sample Variables for details

For full details on setting up the default items to be saved see the Listener Default Configuration documentation. For details of
output files, see the CSV log format or the XML log format.

The entries in jmeter.properties are used to define the defaults; these can be overriden for individual listeners by using the Configure button, as shown below. The settings in jmeter.properties also apply to the listener that is added by using the -l command-line flag.

The figure below shows an example of the result file configuration panel

Result file configuration panel

Parameters

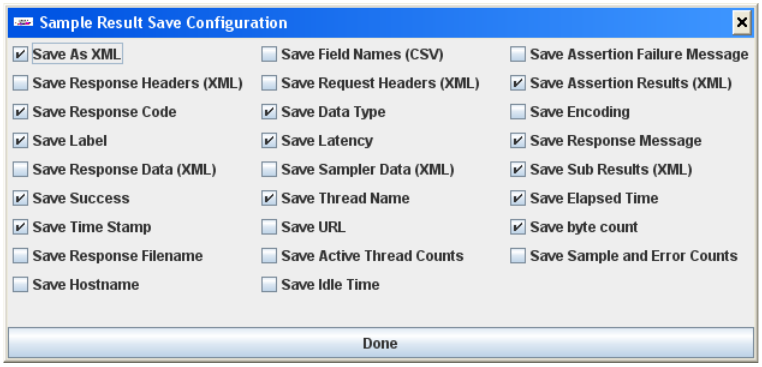
Attribute	Description
Filename	Name of the file containing sample results. The file name can be specified using either a relative or an absolute pat Relative paths are resolved relative to the current working directory (which defaults to the bin/ directory). Version after 2.4 also support paths relative to the directory containing the current test plan (JMX file). If the path name be "~/" (or whatever is in the jmeter.save.saveservice.base_prefix JMeter property), then the path is assumed to be rel JMX file location.
Browse...	File Browse Button
Errors	Select this to write/read only results with errors
Successes	Select this to write/read only results without errors. If neither Errors nor Successes is selected, then all results are p
Configure	Configure Button, see below

18.3.1 Sample Result Save Configuration

Listeners can be configured to save different items to the result log files (JTL) by using the Config popup as shown below. Th
as described in the Listener Default Configuration documentation. Items with (CSV) after the name only apply to the CSV for
(XML) only apply to XML format. CSV format cannot currently be used to save any items that include line-breaks.

Note that cookies, method and the query string are saved as part of the "Sampler Data" option.

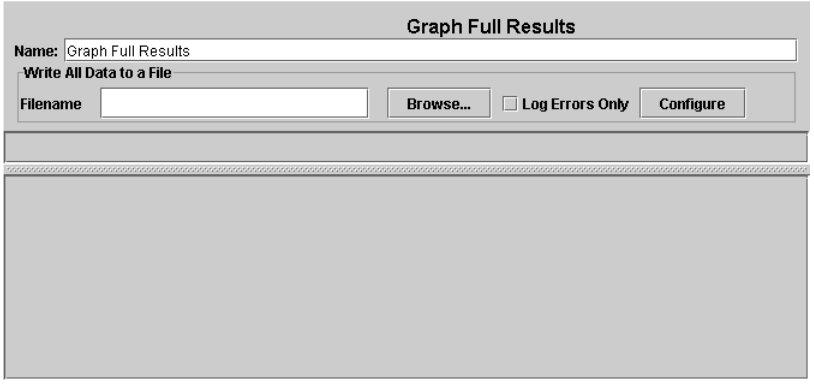
Control Panel



18.3.2 Graph Full Results

No Description

Control Panel



18.3.3 Graph Results

The Graph Results listener generates a simple graph that plots all sample times. Along the bottom of the graph, the current sar current average of all samples(blue), the current standard deviation (red), and the current throughput rate (green) are displayec

The throughput number represents the actual number of requests/minute the server handled. This calculation includes any dela test and JMeter's own internal processing time. The advantage of doing the calculation like this is that this number represents server in fact handled that many requests per minute, and you can increase the number of threads and/or decrease the delays tc maximum throughput. Whereas if you made calculations that factored out delays and JMeter's processing, it would be unclear conclude from that number.

Control Panel

The following table briefly describes the items on the graph. Further details on the precise meaning of the statistical terms can
e.g. Wikipedia - or by consulting a book on statistics.

- Data - plot the actual data values
- Average - plot the Average
- Median - plot the Median (midway value)
- Deviation - plot the Standard Deviation (a measure of the variation)
- Throughput - plot the number of samples per unit of time

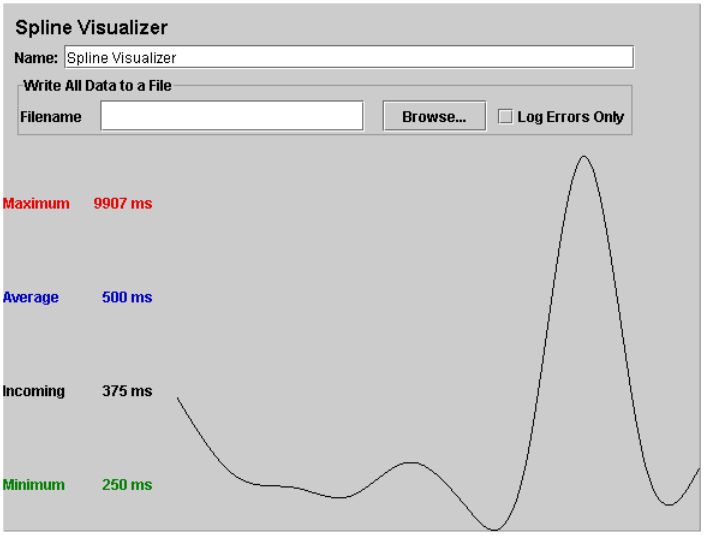
The individual figures at the bottom of the display are the current values. "Latest Sample" is the current elapsed sample time,
"Data".

18.3.4 Spline Visualizer

The Spline Visualizer provides a view of all sample times from the start of the test till the end, regardless of how many sampl
The spline has 10 points, each representing 10% of the samples, and connected using spline logic to show a single continuous

The graph is automatically scaled to fit within the window. This needs to be borne in mind when comparing graphs.

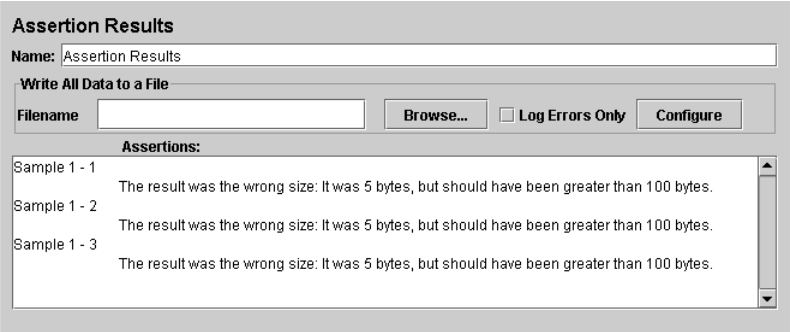
Control Panel



18.3.5 Assertion Results

The Assertion Results visualizer shows the Label of each sample taken. It also reports failures of any Assertions that are part

Control Panel



See Also:

- Response Assertion

18.3.6 View Results Tree

The View Results Tree shows a tree of all sample responses, allowing you to view the response for any sample. In addition to you can see the time it took to get this response, and some response codes. Note that the Request panel only shows the header does not show any headers (such as Host) that may be added by the HTTP protocol implementation.

There are several ways to view the response, selectable by a drop-down box at the bottom of the left hand panel.

- HTML
- HTML (download embedded resources)
- JSON
- Regexp Tester
- Text
- XML

Additional renderers can be created. The class must implement the interface `org.apache.jmeter.visualizers.ResultRen` the abstract class `org.apache.jmeter.visualizers.SamplerResultTab` , and the compiled code must be available to JMe the lib/ext directory).

The default "Text" view shows all of the text contained in the response. Note that this will only work if the response content-t text. If the content-type begins with any of the following, it is considered as binary, otherwise it is considered to be text.

```
image/
audio/
video/
```

If there is no content-type provided, then the content will not be displayed in the any of the Response Data panels. You can us file to save the data in this case. Note that the response data will still be available in the sample result, so can still be accessed

If the response data is larger than 200K, then it won't be displayed. To change this limit, set the JMeter property **view.results.** can also use save the entire response to a file using Save Responses to a file .

The HTML view attempts to render the response as HTML. The rendered HTML is likely to compare poorly to the view one browser; however, it does provide a quick approximation that is helpful for initial result evaluation. No images etc are downlo (download embedded resources) option is selected, the renderer may download images and style-sheets etc referenced by the

The XML view will show response in tree style. Any DTD nodes or Prolog nodes will not show up in tree; however, response nodes.

The JSON view will show the response in tree style (also handles JSON embedded in JavaScript).

Most of the views also allow the displayed data to be searched; the result of the search will be high-lighted in the display abov Control panel screenshot below shows one result of searching for "Java". Note that the search operates on the visible text, so y results when searching the Text and HTML views.

The "Regexp Tester" view only works for text responses. It shows the plain text in the upper panel. The "Test" button allows t Regular Expression to the upper panel and the results will be displayed in the lower panel. For example, the RE **(JMeter\w*).** current JMeter home page gives the following output:

```
Match count: 26
Match[1][0]=JMeter - Apache JMeter</title>
Match[1][1]=JMeter
Match[2][0]=JMeter" title="JMeter" border="0"/></a>
Match[2][1]=JMeter
Match[3][0]=JMeterCommitters">Contributors</a>
Match[3][1]=JMeterCommitters
... and so on ...
```

The first number in [] is the match number; the second number is the group. Group [0] is whatever matched the whole RE. Gr matched the 1st group, i.e. (JMeter\w*) in this case. See Figure 9b (below).

Control Panel

The Control Panel (above) shows an example of an HTML display. Figure 9 (below) shows an example of an XML display.

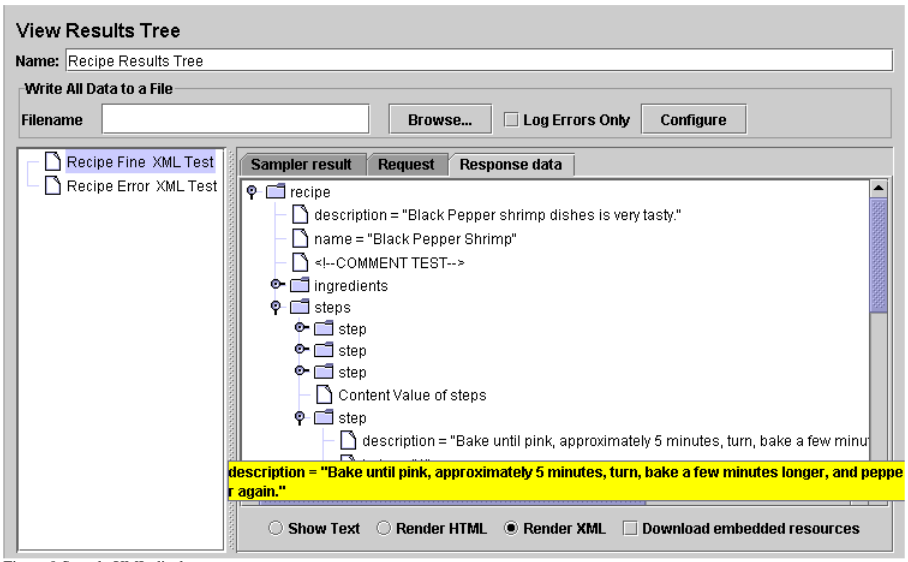


Figure 9 Sample XML display

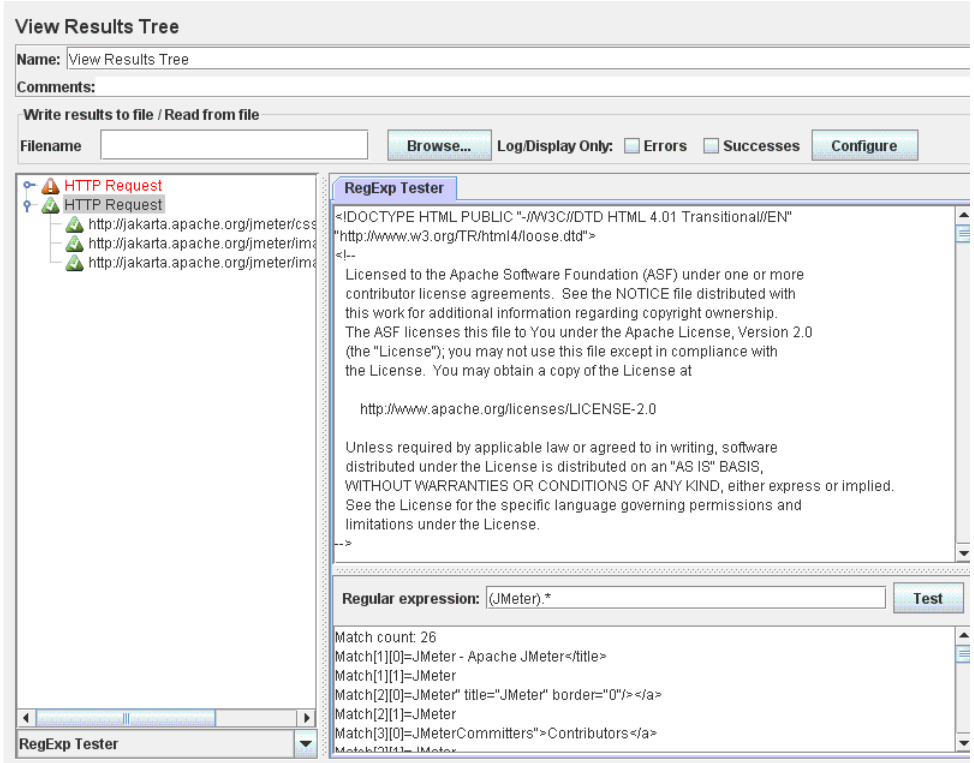


Figure 9a Sample Regexp Test display

18.3.7 Aggregate Report

The aggregate report creates a table row for each differently named request in your test. For each request, it totals the response provides request count, min, max, average, error rate, approximate throughput (request/second) and Kilobytes per second thro is done, the throughput is the actual through for the duration of the entire test.

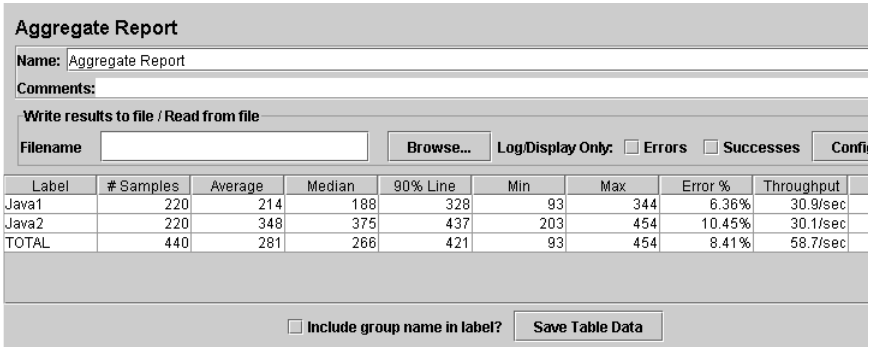
The thoughput is calculated from the point of view of the sampler target (e.g. the remote server in the case of HTTP samples). account the total time over which the requests have been generated. If other samplers and timers are in the same thread, these time, and therefore reduce the throughput value. So two identical samplers with different names will have half the throughput the same name. It is important to choose the sampler names correctly to get the best results from the Aggregate Report.

Calculation of the Median and 90% Line (90 th percentile) values requires additional memory. For JMeter 2.3.4 and earlier, de were saved separately, which meant a lot of memory was needed. JMeter now combines samples with the same elapsed time, used. However, for samples that take more than a few seconds, the probability is that fewer samples will have identical times, memory will be needed. See the Summary Report for a similar Listener that does not store individual samples and so needs co

- Label - The label of the sample. If "Include group name in label?" is selected, then the name of the thread group is adde allows identical labels from different thread groups to be collated separately if required.
- # Samples - The number of samples with the same label
- Average - The average time of a set of results
- Median - The median is the time in the middle of a set of results. 50% of the samples took no more than this time; the re as long.
- 90% Line - 90% of the samples took no more than this time. The remaining samples at least as long as this. (90 th percer
- Min - The shortest time for the samples with the same label
- Max - The longest time for the samples with the same label
- Error % - Percent of requests with errors
- Throughput - the Throughput is measured in requests per second/minute/hour. The time unit is chosen so that the displa When the throughput is saved to a CSV file, it is expressed in requests/second, i.e. 30.0 requests/minute is saved as 0.5.
- Kb/sec - The throughput measured in Kilobytes per second

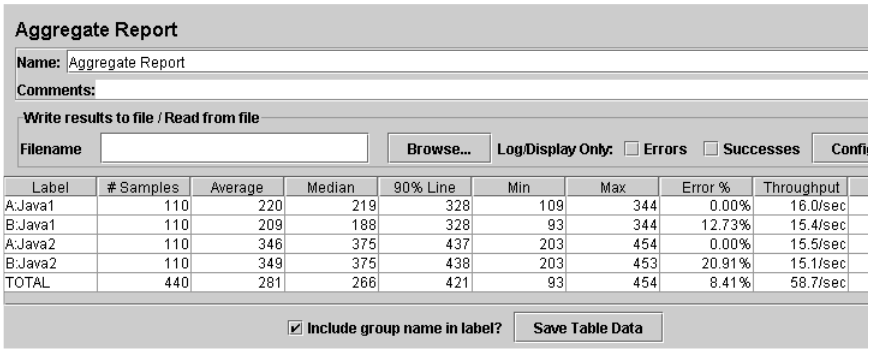
Times are in milliseconds.

Control Panel



Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput
Java1	220	214	188	328	93	344	6.36%	30.9/sec
Java2	220	348	375	437	203	454	10.45%	30.1/sec
TOTAL	440	281	266	421	93	454	8.41%	58.7/sec

The figure below shows an example of selecting the "Include group name" checkbox.



Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput
A:Java1	110	220	219	328	109	344	0.00%	16.0/sec
B:Java1	110	209	188	328	93	344	12.73%	15.4/sec
A:Java2	110	346	375	437	203	454	0.00%	15.5/sec
B:Java2	110	349	375	438	203	453	20.91%	15.1/sec
TOTAL	440	281	266	421	93	454	8.41%	58.7/sec

Sample "Include group name" display

18.3.8 View Results in Table

This visualizer creates a row for every sample result. Like the View Results Tree , this visualizer uses a lot of memory.

Control Panel

View Results in Table

Name: View Results in Table

Comments:

Write All Data to a File / Read from file

Filename Browse... Log Errors Only Configure

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes
1	01:24:03.968	Thread Group 1-1	HTTP 1	1094		12407
2	01:24:04.328	Thread Group 1-2	HTTP 1	1203		12407
3	01:24:04.687	Thread Group 1-3	HTTP 1	1047		12407
4	01:24:05.062	Thread Group 1-1	HTTP 2	1188		2685
5	01:24:05.531	Thread Group 1-2	HTTP 2	1140		2685
6	01:24:05.734	Thread Group 1-3	HTTP 2	1078		2685
7	01:24:06.250	Thread Group 1-1	HTTP 1	1078		12407
8	01:24:06.671	Thread Group 1-2	HTTP 1	1250		12407
9	01:24:06.812	Thread Group 1-3	HTTP 1	1141		12407
10	01:24:07.328	Thread Group 1-1	HTTP 2	1140		2685
11	01:24:07.953	Thread Group 1-3	HTTP 2	1000		2685
12	01:24:07.921	Thread Group 1-2	HTTP 2	1235		2685
13	01:24:08.484	Thread Group 1-1	HTTP 1	1016		12407
14	01:24:08.953	Thread Group 1-3	HTTP 1	1250		12407
15	01:24:09.156	Thread Group 1-2	HTTP 1	1187		12407
16	01:24:09.500	Thread Group 1-1	HTTP 2	1031		2685
17	01:24:10.203	Thread Group 1-3	HTTP 2	1203		2685
18	01:24:10.343	Thread Group 1-2	HTTP 2	1094		2685
19	01:24:10.531	Thread Group 1-1	HTTP 1	1031		12407
20	01:24:11.406	Thread Group 1-3	HTTP 1	1140		12407
21	01:24:11.562	Thread Group 1-1	HTTP 2	1031		2685
22	01:24:11.437	Thread Group 1-2	HTTP 1	1172		12407
23	01:24:12.546	Thread Group 1-3	HTTP 2	1016		2685
24	01:24:12.609	Thread Group 1-1	HTTP 1	1062		12407
25	01:24:12.609	Thread Group 1-2	HTTP 2	1078		2685
26	01:24:13.562	Thread Group 1-3	HTTP 1	1000		12407
27	01:24:13.671	Thread Group 1-1	HTTP 2	1110		2685
28	01:24:13.687	Thread Group 1-2	HTTP 1	1125		12407
29	01:24:14.562	Thread Group 1-3	HTTP 2	1234		2685
30	01:24:14.812	Thread Group 1-2	HTTP 2	1234		2685

No of Samples 30 Latest Sample 1234 Average 1120 Deviation 79

18.3.9 Simple Data Writer

This listener can record results to a file but not to the UI. It is meant to provide an efficient means of recording data by elimin
When running in non-GUI mode, the -l flag can be used to create a data file. The fields to save are defined by JMeter properti
jmeter.properties file for details.

Control Panel

Simple Data Writer

Name: Simple Data Writer

Comments:

Write results to file / Read from file

Filename Browse... Log/Display Only: Errors Successes Confi

18.3.10 Monitor Results

Monitor Results is a new Visualizer for displaying server status. It is designed for Tomcat 5, but any servlet container can por
use this monitor. There are two primary tabs for the monitor. The first is the "Health" tab, which will show the status of one o
second tab labled "Performance" shows the performance for one server for the last 1000 samples. The equations used for the l
included in the Visualizer.

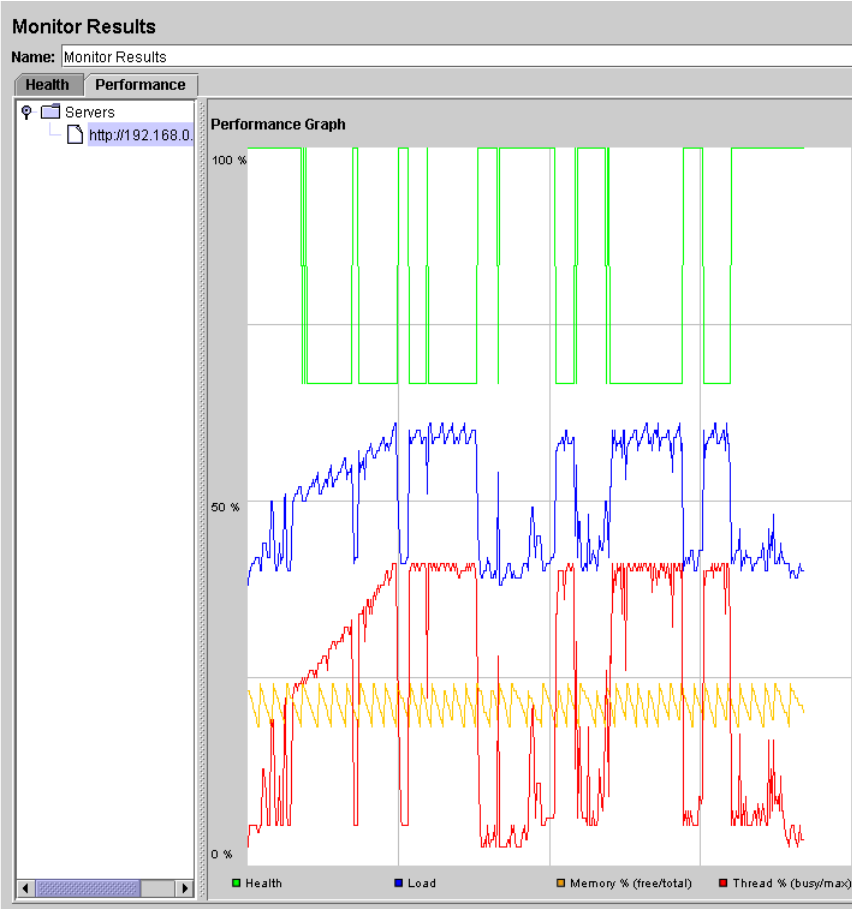
Currently, the primary limitation of the monitor is system memory. A quick benchmark of memory usage indicates a buffer of
100 servers would take roughly 10Mb of RAM. On a 1.4Ghz centrino laptop with 1Gb of ram, the monitor should be able to h
servers.

As a general rule, monitoring production systems should take care to set an appropriate interval. Intervals shorter than 5 secon
and have a potential of impacting the server. With a buffer of 1000 data points at 5 second intervals, the monitor would check
times a minute or 720 times a hour. This means the buffer shows the performance history of each machine for the last hour.

The monitor requires Tomcat 5 or above. Use a browser to check that you can access the Tomcat status servlet OK.

For a detailed description of how to use the monitor, please refer to Building a Monitor Test Plan

Control Panel



18.3.11 Distribution Graph (alpha)

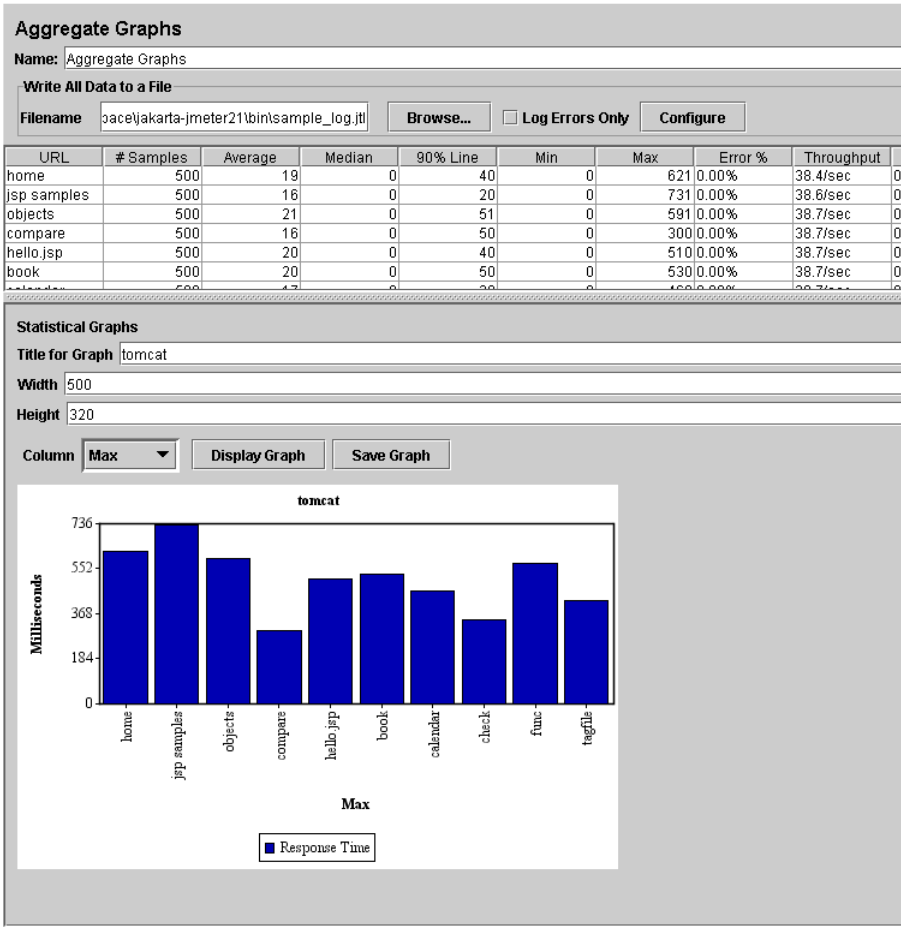
The distribution graph will display a bar for every unique response time. Since the granularity of System.currentTimeMillis() 90% threshold should be within the width of the graph. The graph will draw two threshold lines: 50% and 90%. What this me response times finished between 0 and the line. The same is true of 90% line. Several tests with Tomcat were performed using requests. The graph was able to display the distribution without any problems and both the 50% and 90% line were within the performant application will generally produce results that clump together. A poorly written application that has memory leaks fluctuations. In those situations, the threshold lines may be beyond the width of the graph. The recommended solution to this the webapp so it performs well. If your test plan produces distribution graphs with no apparent clumping or pattern, it may ind The only way to know for sure is to use a profiling tool.

Control Panel

18.3.12 Aggregate Graph

The aggregate graph is similar to the aggregate report. The primary difference is the aggregate graph provides an easy way to and save the graph as a PNG file. By default, the aggregate graph will generate a bar chart 450 x 250 pixels.

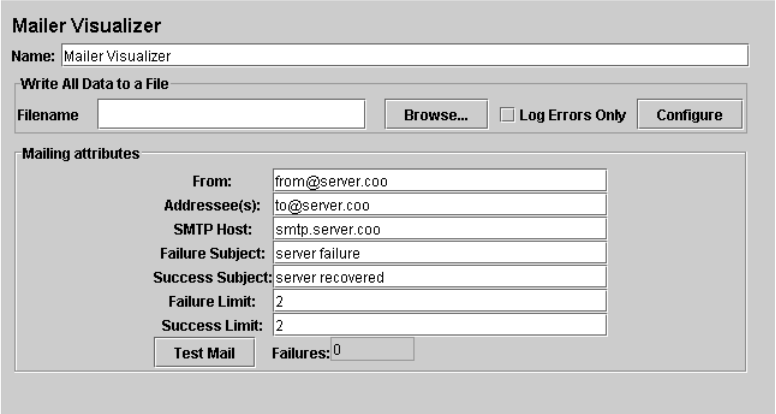
Control Panel



18.3.13 Mailer Visualizer

The mailer visualizer can be set up to send email if a test run receives too many failed responses from the server.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
From	Email address to send messages from.
Addressee(s)	Email address to send messages to, comma-separated.
SMTP Host	IP address or host name of SMTP (email redirector) server.
Failure Subject	Email subject line for fail messages.
Success Subject	Email subject line for success messages.
Failure Limit	Once this number of failed responses is exceeded, a failure email is sent - i.e. set the count to 0 to send an e-ma first failure.
Success Limit	Once this number of successful responses is exceeded **after previously reaching the failure limit** , a success e sent. The mailer will thus only send out messages in a sequence of failed-succeeded-failed-succeeded, etc.
Test Mail	Press this button to send a test mail
Failures	A field that keeps a running total of number of failures so far received.

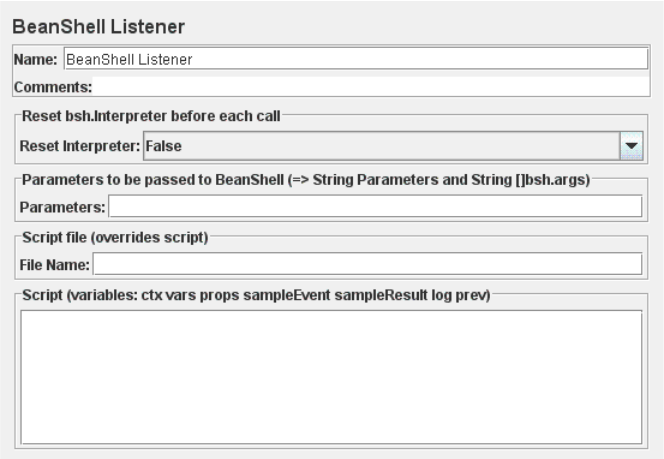
18.3.14 BeanShell Listener

The BeanShell Listener allows the use of BeanShell for processing samples for saving etc.

For full details on using BeanShell, please see the BeanShell website.

The test element supports the ThreadListener and TestListener methods. These should be defined in the initialisation file. See BeanShellListeners.bshrc for example definitions.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree. The name is stored in the script variable Label
Reset bsh.Interpreter before each call	If this option is selected, then the interpreter will be recreated for each sample. This may be necessary for some long running scripts. For further information, see Best Practices - BeanShell scripting .
Parameters	Parameters to pass to the BeanShell script. The parameters are stored in the following variables: • Parameters - string containing the parameters as a single variable • bsh.args - String array containing parameters, split on white-space
Script file	A file containing the BeanShell script to run. The file name is stored in the script variable FileName
Script	The BeanShell script to run. The return value is ignored.

Before invoking the script, some variables are set up in the BeanShell interpreter:

- log - (Logger) - can be used to write to the log file
- ctx - (JMeterContext) - gives access to the context
- vars - (JMeterVariables) - gives read/write access to variables: vars.get(key); vars.put(key,val); vars.putObject("OBJ1",
- props - (JMeterProperties) - e.g. props.get("START.HMS"); props.put("PROP1","1234");
- sampleResult, prev - (SampleResult) - gives access to the previous SampleResult
- sampleEvent (SampleEvent) gives access to the current sample event

For details of all the methods available on each of the above variables, please check the Javadoc

If the property **beanshell.listener.init** is defined, this is used to load an initialisation file, which can be used to define methods BeanShell script.

18.3.15 Summary Report

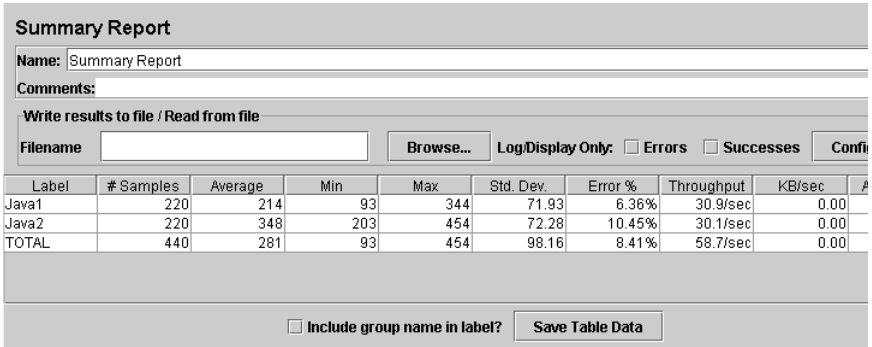
The summary report creates a table row for each differently named request in your test. This is similar to the Aggregate Repor less memory.

The thoughput is calculated from the point of view of the sampler target (e.g. the remote server in the case of HTTP samples). account the total time over which the requests have been generated. If other samplers and timers are in the same thread, these time, and therefore reduce the throughput value. So two identical samplers with different names will have half the throughput the same name. It is important to choose the sampler labels correctly to get the best results from the Report.

- Label - The label of the sample. If "Include group name in label?" is selected, then the name of the thread group is adde allows identical labels from different thread groups to be collated separately if required.
- # Samples - The number of samples with the same label
- Average - The average elapsed time of a set of results
- Min - The lowest elapsed time for the samples with the same label
- Max - The longest elapsed time for the samples with the same label
- Std. Dev. - the Standard Deviation of the sample elapsed time
- Error % - Percent of requests with errors
- Throughput - the Throughput is measured in requests per second/minute/hour. The time unit is chosen so that the displa When the throughput is saved to a CSV file, it is expressed in requests/second, i.e. 30.0 requests/minute is saved as 0.5.
- Kb/sec - The throughput measured in Kilobytes per second
- Avg. Bytes - average size of the sample response in bytes. (in JMeter 2.2 it wrongly showed the value in kB)

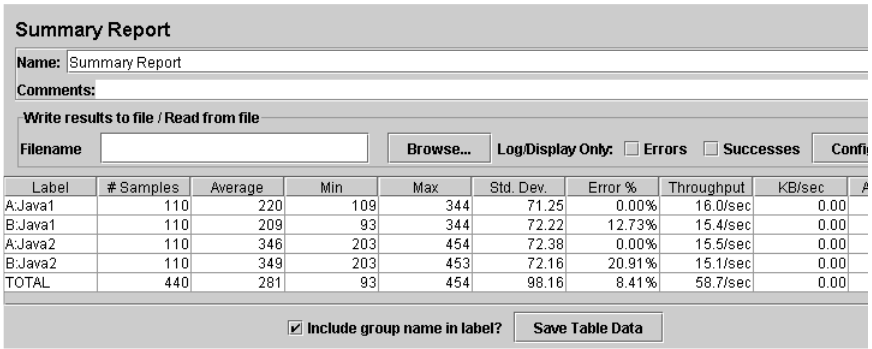
Times are in milliseconds.

Control Panel



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec
Java1	220	214	93	344	71.93	6.36%	30.9/sec	0.00
Java2	220	348	203	454	72.28	10.45%	30.1/sec	0.00
TOTAL	440	281	93	454	98.16	8.41%	58.7/sec	0.00

The figure below shows an example of selecting the "Include group name" checkbox.



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec
A:Java1	110	220	109	344	71.25	0.00%	16.0/sec	0.00
B:Java1	110	209	93	344	72.22	12.73%	15.4/sec	0.00
A:Java2	110	346	203	454	72.38	0.00%	15.5/sec	0.00
B:Java2	110	349	203	453	72.16	20.91%	15.1/sec	0.00
TOTAL	440	281	93	454	98.16	8.41%	58.7/sec	0.00

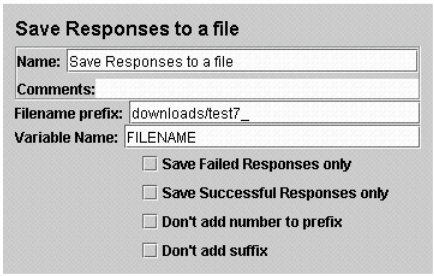
Sample "Include group name" display

18.3.16 Save Responses to a file

This test element can be placed anywhere in the test plan. For each sample in its scope, it will create a file of the response Dat
this is in creating functional tests, but it can also be useful where the response is too large to be displayed in the View Results
name is created from the specified prefix, plus a number (unless this is disabled, see below). The file extension is created from
known. If not known, the file extension is set to 'unknown'. If numbering is disabled, and adding a suffix is disabled, then the
the entire file name. This allows a fixed file name to be generated if required. The generated file name is stored in the sample
saved in the test log output file if required.

The current sample is saved first, followed by any sub-samples (child samples). If a variable name is provided, then the names
in the order that the sub-samples appear. See below.

Control Panel



Parameters

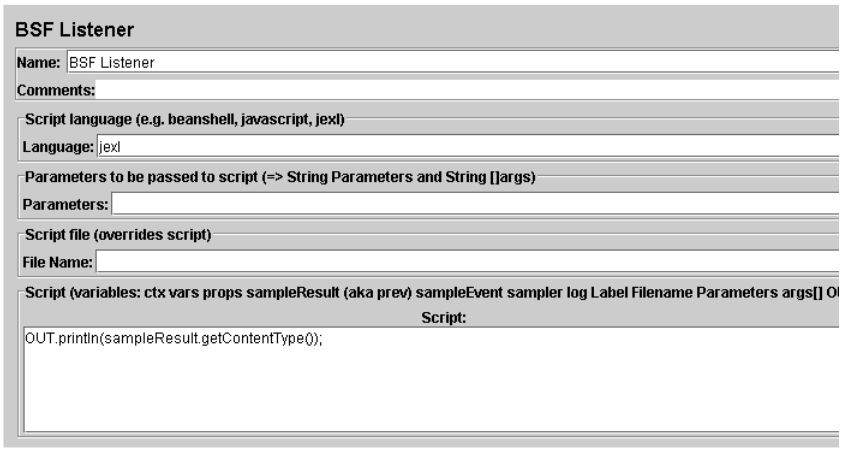
Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Filename Prefix	Prefix for the generated file names; this can include a directory name. Relative paths are resolved relative to t working directory (which defaults to the bin/ directory). Versions of JMeter after 2.4 also support paths relati directory containing the current test plan (JMX file). If the path name begins with "~/" (or whatever is in the jmeter.save.saveservice.base_prefix JMeter property), then the path is assumed to be relative to the JMX file
Variable Name	Name of a variable in which to save the generated file name (so it can be used later in the test plan). If there a samples then a numeric suffix is added to the variable name. E.g. if the variable name is FILENAME, then th sample file name is saved in the variable FILENAME, and the filenames for the child samplers are saved in FILENAME1, FILENAME2 etc.

Save Failed Responses only	If selected, then only failed responses are saved
Save Successful Responses only	If selected, then only successful responses are saved
Don't add number to prefix	If selected, then no number is added to the prefix. If you select this option, make sure that the prefix is unique may be overwritten.
Don't add suffix	If selected, then no suffix is added. If you select this option, make sure that the prefix is unique or the file ma overwritten.

18.3.17 BSF Listener

The BSF Listener allows BSF script code to be applied to sample results.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this element that is shown in the tree.	No
Language	The BSF language to be used	Yes
Parameters	Parameters to pass to the script. The parameters are stored in the following variables: • Parameters - string containing the parameters as a single variable • args - String array containing parameters, split on white-space	No
Script file	A file containing the script to run.	No
Script	The script to run.	Yes (unless script file is pr

The script (or file) is processed using the BSFEngine.exec() method, which does not return a value.

Before invoking the script, some variables are set up. Note that these are BSF variables - i.e. they can be used directly in the s

- log - (Logger) - can be used to write to the log file
- Label - the String Label
- Filename - the script file name (if any)
- Parameters - the parameters (as a String)
- args[] - the parameters as a String array (split on whitespace)
- ctx - (JMeterContext) - gives access to the context
- vars - (JMeterVariables) - gives read/write access to variables: vars.get(key); vars.put(key,val); vars.putObject("OBJ1", vars.getObject("OBJ2");
- props - (JMeterProperties) - e.g. props.get("START.HMS"); props.put("PROP1","1234");
- sampleResult, prev - (SampleResult) - gives access to the SampleResult
- sampleEvent - (SampleEvent) - gives access to the SampleEvent
- sampler - (Sampler)- gives access to the last sampler
- OUT - System.out - e.g. OUT.println("message")

For details of all the methods available on each of the above variables, please check the Javadoc

---

18.3.18.1 JSR223 Listener

The JSR223 Listener allows JSR223 script code to be applied to sample results. For details, see BSF Listener .

18.3.18 Generate Summary Results

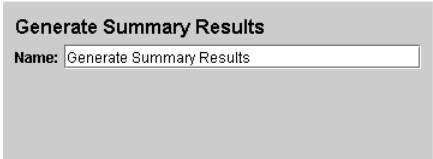
This test element can be placed anywhere in the test plan. Generates a summary of the test run so far to the log file and/or stan running and differential totals are shown. Output is generated every n seconds (default 3 minutes) on the appropriate time bou test runs on the same time will be synchronised. The interval is defined by the property "summariser.interval" - see jmeter.pro mainly intended for batch (non-GUI) runs. The output looks like the following:

```
label +   171 in  20.3s =    8.4/s Avg:  1129 Min:  1000 Max:  1250 Err:     0 (0.00%)
label +   263 in  31.3s =    8.4/s Avg:  1138 Min:  1000 Max:  1250 Err:     0 (0.00%)
label =   434 in  50.4s =    8.6/s Avg:  1135 Min:  1000 Max:  1250 Err:     0 (0.00%)
label +   263 in  31.0s =    8.5/s Avg:  1138 Min:  1000 Max:  1250 Err:     0 (0.00%)
label =   697 in  80.3s =    8.7/s Avg:  1136 Min:  1000 Max:  1250 Err:     0 (0.00%)
label +   109 in  12.4s =    8.8/s Avg:  1092 Min:    47 Max:  1250 Err:     0 (0.00%)
label =   806 in  91.6s =    8.8/s Avg:  1130 Min:    47 Max:  1250 Err:     0 (0.00%)
```

The "label" is the the name of the element. The "+" means that the line is a delta line, i.e. shows the changes since the last out that the line is a totals line, i.e. it shows the running total. Entries in the jmeter log file also include time-stamps. The example means that there were 806 samples recorded in 91.6 seconds, and that works out at 8.8 samples per second. The Avg (Average Max(imum) times are in milliseconds. "Err" means number of errors (also shown as percentage). The last two lines will appea They will not be synchronised to the appropriate time boundary. Note that the initial and final deltas may be for less than the i above this is 30 seconds). The first delta will generally be lower, as JMeter synchronises to the interval boundary. The last del test will generally not finish on an exact interval boundary.

The label is used to group sample results together. So if you have multiple Thread Groups and want to summarize across them label - or add the summariser to the Test Plan (so all thread groups are in scope). Different summary groupings can be implem suitable labels and adding the summarisers to appropriate parts of the test plan.

Control Panel



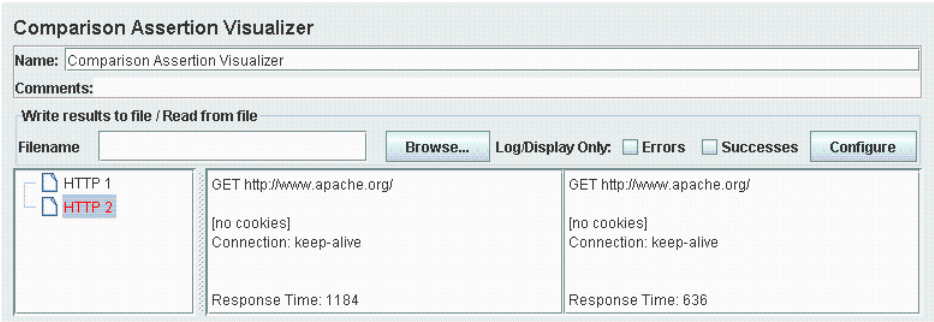
Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree. It appears as the "label" in the output. Details for all el with the same label will be added together.

18.3.19 Comparison Assertion Visualizer

The Comparison Assertion Visualizer shows the results of any Compare Assertion elements.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this element that is shown in the tree.	Yes

^

18.4 Configuration Elements

Configuration elements can be used to set up defaults and variables for later use by samplers. Note that these elements are pro
the scope in which they are found, i.e. before any samplers in the same scope.

18.4.1 CSV Data Set Config

CSV Data Set Config is used to read lines from a file, and split them into variables. It is easier to use than the __CSVRead() a
functions. It is well suited to handling large numbers of variables, and is also useful for tesing with "random" and unique valu
random values at run-time is expensive in terms of CPU and memory, so just create the data in advance of the test. If necessar
from the file can be used in conjunction with a run-time parameter to create different sets of values from each run - e.g. using
is much cheaper than generating everything at run-time.

Versions of JMeter after 2.3.1 allow variables to be quoted; this allows the value to contain a delimiter. Previously it was nece
delimiter that was not used in any values.

Versions of JMeter after 2.3.4 support CSV files which have a header line defining the column names. To enable this, leave th
field empty. The correct delimiter must be provided.

By default, the file is only opened once, and each thread will use a different line from the file. However the order in which lin
threads depends on the order in which they execute, which may vary between iterations. Lines are read at the start of each test
name and mode are resolved in the first iteration.

See the description of the Share mode below for additional options (JMeter 2.3.2+). If you want each thread to have its own se
will need to create a set of files, one for each thread. For example test1.csv, test2.csv,... testn.csv. Use the filename `test${__t`
set the "Sharing mode" to "Current thread".

CSV Dataset variables are defined at the start of each test iteration. As this is after configuration processing is completed, they cannot be used for some configuration items - such as JDBC Config - that process their contents at configuration time (see Bug 40394) However the variables do work in the HTTP Auth Manager, as the username etc are processed at run-time.

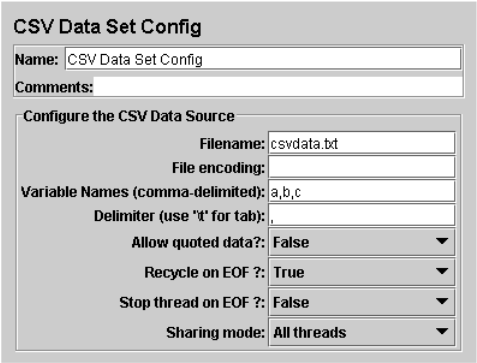
As a special case, the string "\t" (without quotes) in the delimiter field is treated as a Tab.

When the end of file (EOF) is reached, and the recycle option is true, reading starts again with the first line of the file.

If the recycle option is false, and stopThread is false, then all the variables are set to **<EOF>** when the end of file is reached. T
changed by setting the JMeter property **csvdataset.eofstring** .

If the Recycle option is false, and Stop Thread is true, then reaching EOF will cause the thread to be stopped.

Control Panel



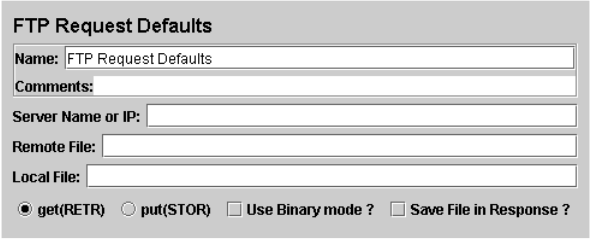
Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Filename	Name of the file to be read. **Relative file names are resolved with respect to the path of the active test plan.** file names are also supported, but note that they are unlikely to work in remote mode, unless the remote server h same directory structure. If the same physical file is referenced in two different ways - e.g. csvdata.txt and ./csvd then these are treated as different files. If the OS does not distinguish between upper and lower case, csvData.TX also be opened separately.
File Encoding	The encoding to be used to read the file, if not the platform default.

Variable Names	List of variable names (comma-delimited). Versions of JMeter after 2.3.4 support CSV header lines: if the varia field empty, then the first line of the file is read and interpreted as the list of column names. The names must be by the delimiter character. They can be quoted using double-quotes.
Delimiter	Delimiter to be used to split the records in the file. If there are fewer values on the line than there are variables tl remaining variables are not updated - so they will retain their previous value (if any).
Allow quoted data?	Should the CSV file allow values to be quoted?
Recycle on EOF?	Should the file be re-read from the beginning on reaching EOF? (default is true)
Stop thread on EOF?	Should the thread be stopped on EOF, if Recycle is false? (default is false)
Sharing mode	• All threads - (the default) the file is shared between all the threads. • Current thread group - each file is opened once for each thread group in which the element appears • Current thread - each file is opened separately for each thread • Identifier - all threads sharing the same identifier share the same file. So for example if you have 4 thread you could use a common id for two or more of the groups to share the file between them. Or you could us thread number to share the file between the same thread numbers in different thread groups.

18.4.2 FTP Request Defaults

Control Panel



18.4.3 HTTP Authorization Manager

If there is more than one Authorization Manager in the scope of a Sampler, there is currently no way to specify which one is to be used.

The Authorization Manager lets you specify one or more user logins for web pages that are restricted using server authenticati of authentication when you use your browser to access a restricted page, and your browser displays a login dialog box. JMeter information when it encounters this type of page.

The Authorisation headers are not shown in the Tree View Listener.

In versions of JMeter after 2.2, the HttpClient sampler defaults to pre-emptive authentication if the setting has not been define the values as below, in which case authentication will only be performed in response to a challenge.

```
jmeter.properties:
httpclient.parameters.file=httpclient.parameters

httpclient.parameters:
http.authentication.preemptive$Boolean=false
```

Note: the above settings only apply to the HttpClient sampler (and the SOAP samplers, which use Httpclient).

When looking for a match against a URL, JMeter checks each entry in turn, and stops when it finds the first match. Thus the most specific URLs should appear first in the list, followed by less specific ones. Duplicate URLs will be ignored. If you want to use different usernames/passwords for different threads, you can use variables. These can be set up using a CSV Data Set Config Element (for example).

Control Panel

Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Base URL	A partial or complete URL that matches one or more HTTP Request URLs. As an example, say you specify a Bas "http://jakarta.apache.org/restricted/" with a username of "jmeter" and a password of "jmeter". If you send an HTT to the URL "http://jakarta.apache.org/restricted/ant/myPage.html", the Authorization Manager sends the login info for the user named, "jmeter".
Username	The username to authorize.
Password	The password for the user.
Domain	The domain to use for NTLM.
Realm	The realm to use for NTLM.

The Realm only applies to the HttpClient sampler. In JMeter 2.2, the domain and realm did not have separate columns, and were encoded as part of the user name in the form: [domain\]username[@realm]. This was an experimental feature and has been removed.

Controls:

- Add Button - Add an entry to the authorization table.
- Delete Button - Delete the currently selected table entry.
- Load Button - Load a previously saved authorization table and add the entries to the existing authorization table entries.
- Save As Button - Save the current authorization table to a file.

When you save the Test Plan, JMeter automatically saves all of the authorization table entries - including any passwords, which are not encrypted.

Authorization Example

Download this example. In this example, we created a Test Plan on a local server that sends three HTTP requests, two requiri is open to everyone. See figure 10 to see the makeup of our Test Plan. On our server, we have a restricted directory named, "s two files, "index.html" and "index2.html". We created a login id named, "kevin", which has a password of "spot". So, in our A we created an entry for the restricted directory and a username and password (see figure 11). The two HTTP requests named " "SecretPage2" make requests to "/secret/index.html" and "/secret/index2.html". The other HTTP request, named "NoSecretPa "/index.html".

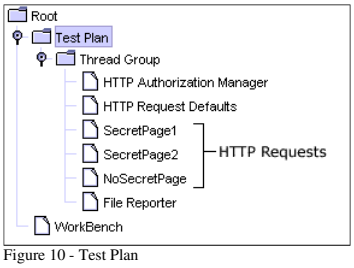


Figure 10 - Test Plan

Figure 11 - Authorization Manager Control Panel

When we run the Test Plan, JMeter looks in the Authorization table for the URL it is requesting. If the Base URL matches the
passes this information along with the request.

You can download the Test Plan, but since it is built as a test for our local server, you will not be able to run it. However, you can use it as a reference in constructing your own Test Plan.

18.4.4 HTTP Cache Manager

This is a new element, and is liable to change

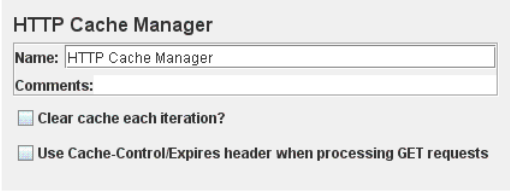
The HTTP Cache Manager is used to add caching functionality to HTTP requests within its scope.

If a sample is successful (i.e. has response code 2xx) then the Last-Modified and Etag (and Expired if relevant) values are sav
executing the next sample, the sampler checks to see if there is an entry in the cache, and if so, the If-Last-Modified and If-Nc
headers are set for the request.

Additionally, if the "Use Cache-Control/Expires header" option is selected, then the Cache-Control/Expires value is checked a
time. If the request is a GET request, and the timestamp is in the future, then the sampler returns immediately, without request
remote server. This is intended to emulate browser behaviour. Note that the Cache-Control header must be "public" and only t
option is processed.

If the requested document has not changed since it was cached, then the response body will be empty. Likewise if the Expires
This may cause problems for Assertions.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree
Clear cache each iteration	If selected, then the cache is cleared at the start of the thre
Use Cache Control/Expires header when processing GET requests	See description above.

18.4.4 HTTP Cookie Manager

If there is more than one Cookie Manager in the scope of a Sampler, there is currently no way to specify which one is to be used. Also, a cookie stored in one cookie manager is not available to any other manager, so use multiple Cookie Managers with care.

The Cookie Manager element has two functions:

First, it stores and sends cookies just like a web browser. If you have an HTTP Request and the response contains a cookie, th
automatically stores that cookie and will use it for all future requests to that particular web site. Each JMeter thread has its ow
area". So, if you are testing a web site that uses a cookie for storing session information, each JMeter thread will have its own
such cookies do not appear on the Cookie Manager display, but they can be seen using the View Results Tree Listener.

JMeter version 2.3.2 and earlier did not check that received cookies were valid for the URL. This meant that cross-domain coo
might be used later. This has been fixed in later versions. To revert to the earlier behaviour, define the JMeter property
"CookieManager.check.cookies=false".

Received Cookies can be stored as JMeter thread variables (versions of JMeter after 2.3.2 no longer do this by default). To sav
define the property "CookieManager.save.cookies=true". Also, cookies names are prefixed with "COOKIE_" before they are s
accidental corruption of local variables) To revert to the original behaviour, define the property "CookieManager.name.prefix=
spaces). If enabled, the value of a cookie with the name TEST can be referred to as ${COOKIE_TEST}.

Second, you can manually add a cookie to the Cookie Manager. However, if you do this, the cookie will be shared by all JMet

Note that such Cookies are created with an Expiration time far in the future

Since version 2.0.3, cookies with null values are ignored by default. This can be changed by setting the JMeter property:
CookieManager.delete_null_cookies=false. Note that this also applies to manually defined cookies - any such cookies will be

display when it is updated. Note also that the cookie name must be unique - if a second cookie is defined with the same name, first.

Control Panel

Parameters

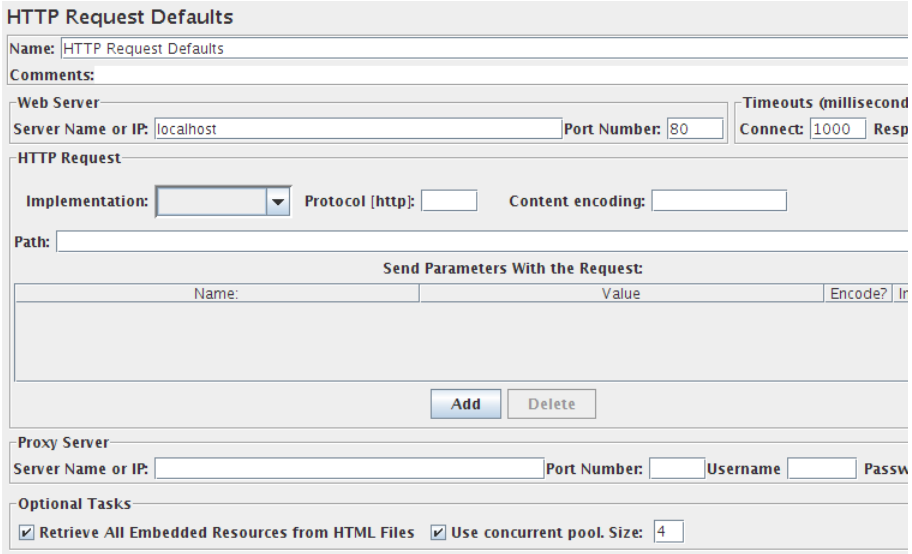
Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Clear Cookies each Iteration	If selected, all server-defined cookies are cleared each time the main Thread Group loop is executed. In JMeter versions after 2.3, any cookies defined in the GUI are not cleared.
Cookie Policy	The cookie policy that will be used to manage the cookies. "compatibility" is the default, and should work in mo cases. See http://jakarta.apache.org/httpcomponents/httpclient-3.x/cookies.html and http://jakarta.apache.org/httpcomponents/httpclient-3.x/apidocs/org/apache/commons/httpclient/cookie/CookiePolicy.html [Note: "ignoreCookies" is equivalent to omitting the CookieManager.]
User-Defined Cookies	This gives you the opportunity to use hardcoded cookies that will be used by all threads during the test executio The "domain" is the hostname of the server (without http://); the port is currently ignored.
Add Button	Add an entry to the cookie table.
Delete Button	Delete the currently selected table entry.
Load Button	Load a previously saved cookie table and add the entries to the existing cookie table entries.
Save As Button	Save the current cookie table to a file (does not save any cookies extracted from HTTP Responses).

18.4.5 HTTP Request Defaults

This element lets you set default values that your HTTP Request controllers use. For example, if you are creating a Test Plan controllers and all of the requests are being sent to the same server, you could add a single HTTP Request Defaults element w or IP" field filled in. Then, when you add the 25 HTTP Request controllers, leave the "Server Name or IP" field empty. The co this field value from the HTTP Request Defaults element.

> In JMeter 2.2 and earlier, port 80 was treated specially - it was ignored if the sampler used the https protocol. JMeter 2.3 and later treat all port values equally; a sampler that does not specify a port will use the HTTP Request Defaults port, if one is provided.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this controller that is shown in the tree.
Server	Domain name or IP address of the web server. e.g. www.example.com. [Do not include the http:// pre
Port	Port the web server is listening to.
Connect Timeout	Connection Timeout. Number of milliseconds to wait for a connection to open. Requires Java 1.5 or l when using the default Java HTTP implementation.
Response Timeout	Response Timeout. Number of milliseconds to wait for a response. Requires Java 1.5 or later when us the default Java HTTP implementation.
Implementation	Java, HttpClient3.1, HttpClient4. If not specified the default depends on the value of the JMeter prope `jmeter.httpsampler` , failing that, the Java implementation is used.
Protocol	HTTP or HTTPS.
Method	HTTP GET or HTTP POST.
Path	The path to resource (for example, /servlets/myServlet). If the resource requires query string paramete add them below in the "Send Parameters With the Request" section. Note that the path is the default f full path, not a prefix to be applied to paths specified on the HTTP Request screens.
Send Parameters With the Request	The query string will be generated from the list of parameters you provide. Each parameter has a *nam value* . The query string will be generated in the correct fashion, depending on the choice of "Method' made (ie if you chose GET, the query string will be appended to the URL, if POST, then it will be sen separately). Also, if you are sending a file using a multipart form, the query string will be created usin multipart form specifications.
Server (proxy)	Hostname or IP address of a proxy server to perform request. [Do not include the http:// prefix.]
Port	Port the proxy server is listening to.
Username	(Optional) username for proxy server.
Password	(Optional) password for proxy server.
Retrieve All Embedded Resources from HTML Files	Tell JMeter to parse the HTML file and send HTTP/HTTPS requests for all images, Java applets, JavaScript files, CSSs, etc. referenced in the file.
Use concurrent pool	Use a pool of concurrent connections to get embedded resources.
Size	Pool size for concurrent connections used to get embedded resources.

Note: radio buttons only have two states - on or off. This makes it impossible to override settings consistently - does off mean off, or does it mean use the current default? JMeter uses the latter (otherwise defaults would not work at all). So if the button is off, then a later element can set it on, but if the button is on, a later element cannot set it off.

18.4.6 HTTP Header Manager

The Header Manager lets you add or override HTTP request headers.

Versions of JMeter up to 2.3.2 supported only one Header Manager per sampler; if there were more in scope, then only the las

JMeter now supports multiple Header Managers . The header entries are merged to form the list for the sampler. If an entr matches an existing header name, it replaces the previous entry, unless the entry value is empty, in which case any existing en allows one to set up a default set of headers, and apply adjustments to particular samplers.

Control Panel

Parameters

Attribute	Description	
Name	Descriptive name for this element that is shown in the tree.	No
Name (Header)	Name of the request header. Two common request headers you may want to experiment with are "User-Agent" and "Referer".	No (You sho however)
Value	Request header value.	No (You sho however)
Add Button	Add an entry to the header table.	N/A
Delete Button	Delete the currently selected table entry.	N/A
Load Button	Load a previously saved header table and add the entries to the existing header table entries.	N/A
Save As Button	Save the current header table to a file.	N/A

Header Manager example

Download this example. In this example, we created a Test Plan that tells JMeter to override the default "User-Agent" request particular Internet Explorer agent string instead. (see figures 9 and 10).

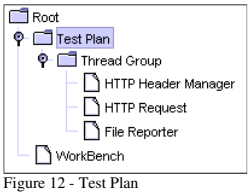


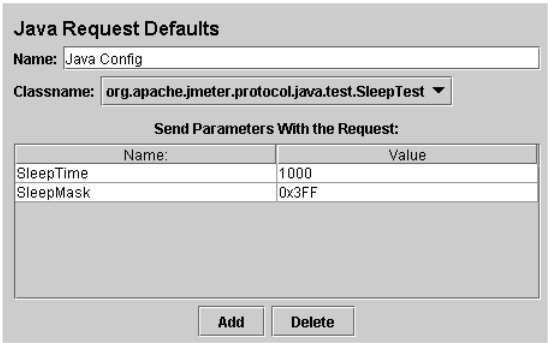
Figure 12 - Test Plan

Figure 13 - Header Manager Control Panel

18.4.7 Java Request Defaults

The Java Request Defaults component lets you set default values for Java testing. See the Java Request .

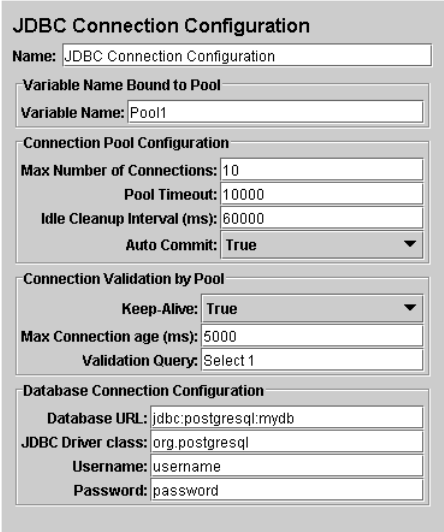
Control Panel



18.4.8 JDBC Connection Configuration

Creates a database connection (used by JDBC Request Sampler) from the supplied JDBC Connection settings. The connectior pooled between threads. Otherwise each thread gets its own connection. The connection configuration name is used by the JD the appropriate connection.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for the connection configuration that is shown in the tree.
Variable Name	The name of the variable the connection is tied to. Multiple connections can be used, each tied to a differen allowing JDBC Samplers to select the appropriate connection. **Each name must be different. If there are configuration elements using the same name, only one will be saved. JMeter versions after 2.3 log a n duplicate name is detected.**
Max Number of Connections	Maximum number of connections allowed in the pool. In most cases, **set this to zero (0)** . This means that will get its own pool with a single connection in it, i.e. the connections are not shared betweeen threads. If you really want to use shared pooling (why?), then set the max count to the same as the number of thread threads don't wait on each other.
Pool timeout	Pool throws an error if the timeout period is exceeded in the process of trying to retrieve a connection
Idle Cleanup Interval (ms)	Uncertain what exactly this does.
Auto Commit	Turn auto commit on or off for the connections.
Keep-alive	Uncertain what exactly this does.
Max Connection Age (ms)	Uncertain what exactly this does.
Validation Query	A simple query used to determine if the database is still responding.
Database URL	JDBC Connection string for the database.
JDBC Driver class	Fully qualified name of driver class. (Must be in JMeter's classpath - easiest to copy .jar file into JMeter's /l directory).
Username	Name of user to connect as.
Password	Password to connect with.

Different databases and JDBC drivers require different JDBC settings. The Database URL and JDBC Driver class are defined JDBC implementation.

Some possible settings are shown below. Please check the exact details in the JDBC driver documentation.

If JMeter reports **No suitable driver** , then this could mean either:

- The driver class was not found. In this case, there will be a log message such as `DataSourceElement: Could not loa {classname} java.lang.ClassNotFoundException: {classname}`
- The driver class was found, but the class does not support the connection string. This could be because of a syntax error string, or because the the wrong classname was used.

If the database server is not running or is not accessible, then JMeter will report a **java.net.ConnectException** .

Database	Driver class	Database URL
MySQL	com.mysql.jdbc.Driver	jdbc:mysql://host[:port]/dbname
PostgreSQL	org.postgresql.Driver	jdbc:postgresql:{dbname}

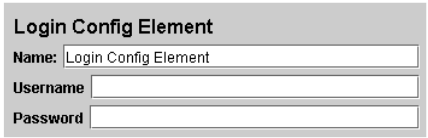
Oracle	oracle.jdbc.OracleDriver	jdbc:oracle:thin:@//host:port/service OR jdbc:oracle:thin:@(description=(address=(host={mc-name})(proto (connect_data=(sid={sid})))
Ingres (2006)	ingres.jdbc.IngresDriver	jdbc:ingres://host:port/db[;attr=value]
SQL Server (MS JDBC driver)	com.microsoft.sqlserver.jdbc.SQLServerDriver	jdbc:sqlserver://host:port;DatabaseName=dbname
Apache Derby	org.apache.derby.jdbc.ClientDriver	jdbc:derby://server[:port]/databaseName[;URLAttributes=value[;...

The above may not be correct - please check the relevant JDBC driver documentation.

18.4.9 Login Config Element

The Login Config Element lets you add or override username and password settings in samplers that use username and passw setup.

Control Panel



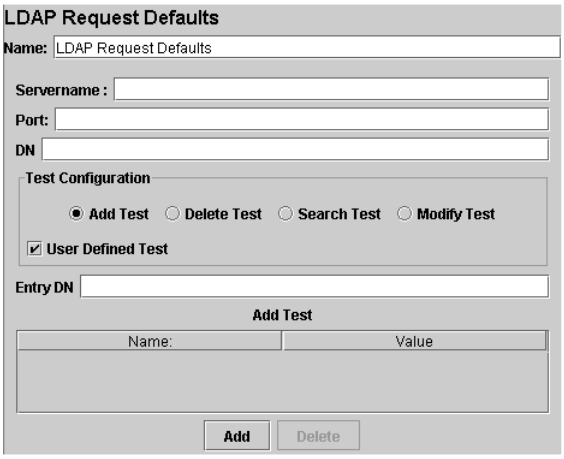
Parameters

Attribute	Description	Required
Name	Descriptive name for this element that is shown in the tree.	No
Username	The default username to use.	No
Password	The default password to use.	No

18.4.10 LDAP Request Defaults

The LDAP Request Defaults component lets you set default values for LDAP testing. See the LDAP Request .

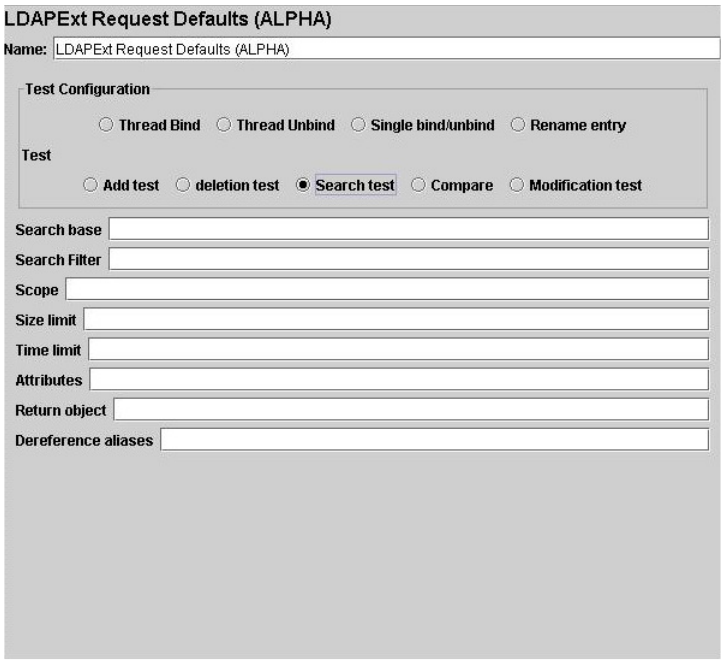
Control Panel



18.4.11 LDAP Extended Request Defaults

The LDAP Extended Request Defaults component lets you set default values for extended LDAP testing. See the LDAP Exte

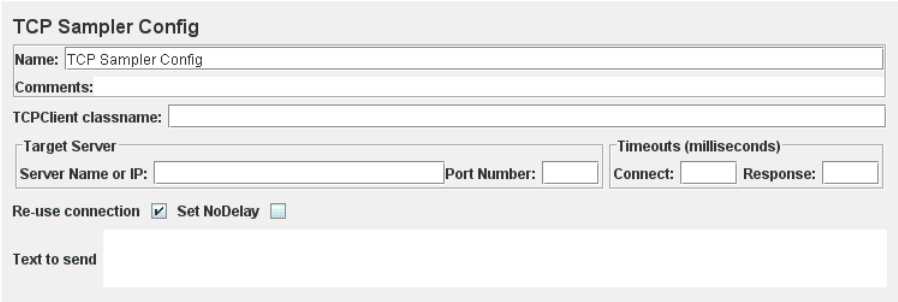
Control Panel



18.4.12 TCP Sampler Config

The TCP Sampler Config provides default data for the TCP Sampler

Control Panel



Parameters

Attribute	Description	Requi
Name	Descriptive name for this element that is shown in the tree.	No
TCPClient classname	Name of the TCPClient class. Defaults to the property tcp.handler, failing that TCPClientImpl.	No
ServerName or IP	Name or IP of TCP server	No
Port Number	Port to be used	No
Re-use connection	If selected, the connection is kept open. Otherwise it is closed when the data has been read.	Yes
Connect Timeout	Connect Timeout (milliseconds, 0 disables).	No
Response Timeout	Response Timeout (milliseconds, 0 disables).	No
Set Nodelay	Should the nodelay property be set?	No
Text to Send	Text to be sent	No

18.4.13 User Defined Variables

The User Defined Variables element lets you define an **initial set of variables** , just as in the Test Plan . **Note that all the UD plan - no matter where they are - are processed at the start.** So you cannot reference variables which are defined as part of Post-Processor.

UDVs should not be used with functions that generate different results each time they are called. Only the result of the will be saved in the variable. However, UDVs can be used with functions such as __P(), for example:

```
HOST      ${__P(host,localhost)}
```

which would define the variable "HOST" to have the value of the JMeter property "host", defaulting to "localhost" if not defir

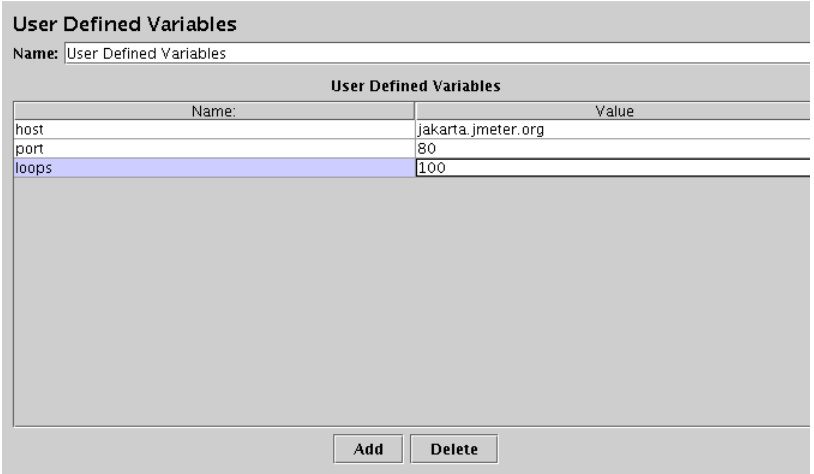
For defining variables during a test run, see User Parameters . UDVs are processed in the order they appear in the Plan, from t

For simplicity, it is suggested that UDVs are placed only at the start of a Thread Group (or perhaps under the Test Plan itself).

Once the Test Plan and all UDVs have been processed, the resulting set of variables is copied to each thread to provide the ini

If a runtime element such as a User Parameters Pre-Processor or Regular Expression Extractor defines a variable with the sam UDV variables, then this will replace the initial value, and all other test elements in the thread will see the updated value.

Control Panel



If you have more than one Thread Group, make sure you use different names for different values, as UDVs are shared between Thread Groups. Also, the variables are not available for use until after the element has been processed, so you cannot reference variables that are defined in the same element. You can reference variables defined in earlier UDVs or on the Test Plan.

Parameters

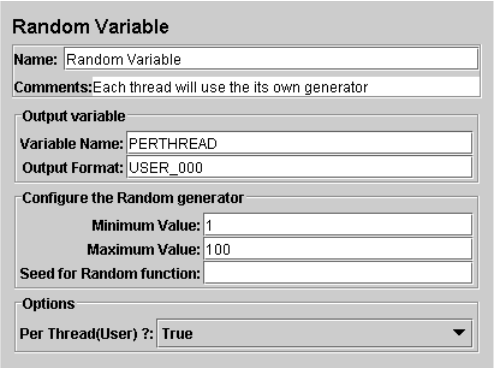
Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
User Defined Variables	Variable name/value pairs. The string under the "Name" column is what you'll need to place inside the brac ${...} constructs to use the variables later on. The whole ${...} will then be replaced by the string in the "Va column.

18.4.14 Random Variable

The Random Variable Config Element is used to generate random numeric strings and store them in variable for use later. It's User Defined Variables together with the __Random() function.

The output variable is constructed by using the random number generator, and then the resulting number is formatted using th number is calculated using the formula `minimum+Random.nextInt(maximum-minimum+1)` . Random.nextInt() requires a pos means that maximum-minimum - i.e. the range - must be less than 2147483647, however the minimum and maximum values so long as the range is OK.

Control Panel



Parameters

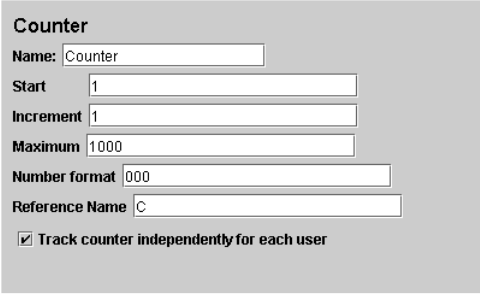
Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Variable Name	The name of the variable in which to store the random string.
Format String	The java.text.DecimalFormat format string to be used. For example "000" which will generate numbers with at digits, or "USER_000" which will generate output of the form USER_nnn. If not specified, the default is to ge number using Long.toString()
Minimum Value	The minimum value (long) of the generated random number.
Maximum Value	The maximum value (long) of the generated random number.
Random Seed	The seed for the random number generator. Default is the current time in milliseconds.
Per Thread (User)?	If False, the generator is shared between all threads in the thread group. If True, then each thread has its own ra generator.

18.4.15 Counter

Allows the user to create a counter that can be referenced anywhere in the Thread Group. The counter config lets the user con a maximum, and the increment. The counter will loop from the start to the max, and then start over with the start, continuing c test is ended.

From version 2.1.2, the counter now uses a long to store the value, so the range is from -2^63 to 2^63-1.

Control Panel



Parameters

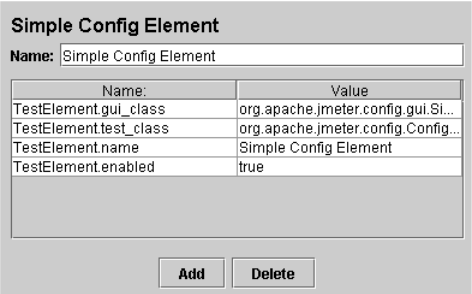
Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Start	The starting number for the counter. The counter will equal this number during the first iteration.
Increment	How much to increment the counter by after each iteration.
Maximum	If the counter exceeds the maximum, then it is reset to the Start value. For versions after 2.2 the de Long.MAX_VALUE (previously it was 0).
Format	Optional format, e.g. 000 will format as 001, 002 etc. This is passed to DecimalFormat, so any vali can be used. If there is a problem interpreting the format, then it is ignored. [The default format is using Long.toString()]
Reference Name	This controls how you refer to this value in other elements. Syntax is as in user-defined values : `${reference_name}` .

Track Counter Independently for each User	In other words, is this a global counter, or does each user get their own counter? If unchecked, the global (ie, user #1 will get value "1", and user #2 will get value "2" on the first iteration). If checke user has an independent counter.

18.4.16 Simple Config Element

The Simple Config Element lets you add or override arbitrary values in samplers. You can choose the name of the value and t Although some adventurous users might find a use for this element, it's here primarily for developers as a basic GUI that they developing new JMeter components.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Parameter Name	The name of each parameter. These values are internal to JMeter's workings and are not generally documente those familiar with the code will know these values.
Parameter Value	The value to apply to that parameter.

---

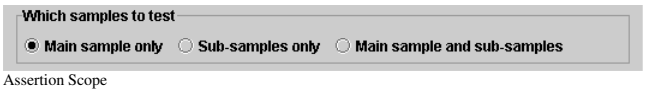
^

18.5 Assertions

Assertions are used to perform additional checks on samplers, and are processed after **every sampler** in the same scope. To er is applied only to a particular sampler, add it as a child of the sampler.

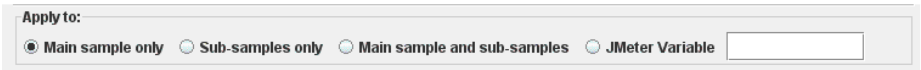
Note: Unless documented otherwise, Assertions are not applied to sub-samples (child samples) - only to the parent sample. In BeanShell Assertions, the script can retrieve sub-samples using the method `prev.getSubResults()` which returns an array o array will be empty if there are none.

Versions of JMeter after 2.3.2 include the option to apply certain assertions to either the main sample, the sub-samples or both apply the assertion to the main sample only. If the Assertion supports this option, then there will be an entry on the GUI which following:



Assertion Scope

or the following



Assertion Scope

If a sub-sampler fails and the main sample is successful, then the main sample will be set to failed status and an Assertion Res JMeter variable option is used, it is assumed to relate to the main sample, and any failure will be applied to the main sample o

The variable **JMeterThread.last_sample_ok** is updated to "true" or "false" after all assertions for a sampler have been run.

18.5.1 Response Assertion

The response assertion control panel lets you add pattern strings to be compared against various fields of the response. The pa

- Contains, Matches: Perl5-style regular expressions
- Equals, Substring: plain text, case-sensitive

A summary of the pattern matching characters can be found at http://jakarta.apache.org/oro/api/org/apache/oro/text/regex/pac

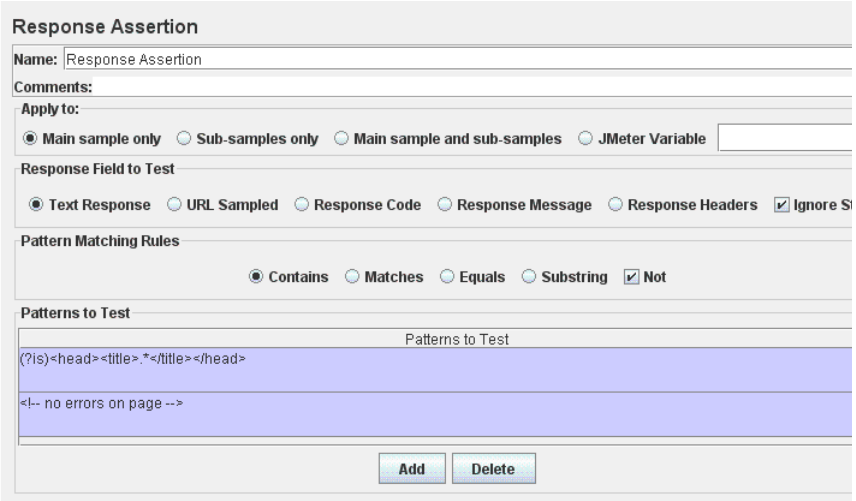
You can also choose whether the strings will be expected to **match** the entire response, or if the response is only expected to **c**
You can attach multiple assertions to any controller for additional flexibility.

Note that the pattern string should not include the enclosing delimiters, i.e. use **Price: \d+** not **/Price: \d+/** .

By default, the pattern is in multi-line mode, which means that the "." meta-character does not match newline. In multi-line m
the start or end of any line anywhere within the string - not just the start and end of the entire string. Note that \s does match n
significant. To override these settings, one can use the *extended regular expression* syntax. For example:

```
        (?i) - ignore case
        (?s) - treat target as single line, i.e. "." matches new-line
        (?is) - both the above
    These can be used anywhere within the expression and remain in effect until overriden.  e.g.
    (?i)apple(?-i) Pie - matches "ApPLe Pie", but not "ApPLe pIe"
    (?s)Apple.+?Pie - matches Apple followed by Pie, which may be on a subsequent line.
    Apple(?s).+?Pie - same as above, but it's probably clearer to use the (?s) at the start.
```

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Apply to:	This is for use with samplers that can generate sub-samples, e.g. HTTP Sampler with embedded resources, Mai samples generated by the Transaction Controller. • Main sample only - assertion only applies to the main sample • Sub-samples only - assertion only applies to the sub-samples • Main sample and sub-samples - assertion applies to both. • JMeter Variable - assertion is to be applied to the contents of the named variable
Response Field to Test	Instructs JMeter which field of the Response to test. • Text Response - the response text from the server, i.e. the body, excluing any HTTP headers. • URL sampled • Response Code - e.g. 200 • Response Message - e.g. OK • Response Headers, including Set-Cookie headers (if any)
Ignore status	Instructs JMeter to set the status to success initially. The overall success of the sample is determined by combining the result of the assertion with the existing Resp When the Ignore Status checkbox is selected, the Response status is forced to successful before evaluating the A HTTP Responses with statuses in the 4xx and 5xx ranges are normally regarded as unsuccessful. The "Ignore st checkbox can be used to set the status successful before performing further checks. Note that this will have the clearing any previous assertion failures, so make sure that this is only set on the first assertion.
Pattern Matching Rules	Indicates how the text being tested is checked against the pattern. • Contains - true if the text contains the regular expression pattern • Matches - true if the whole text matches the regular expression pattern • Equals - true if the whole text equals the pattern string (case-sensitive) • Substring - true if the text contains the pattern string (case-sensitive)

	Equals and Substring patterns are plain strings, not regular expressions. NOT may also be selected to invert the the check.
Patterns to Test	A list of patterns to be tested. Each pattern is tested separately. If a pattern fails, then further patterns are not ch There is no difference between setting up one Assertion with multiple patterns and setting up multiple Assertio pattern each (assuming the other options are the same). **However, when the Ignore Status checkbox is selecte the effect of cancelling any previous assertion failures - so make sure that the Ignore Status checkbox is o on the first Assertion.**

The pattern is a Perl5-style regular expression, but without the enclosing brackets.

Assertion Examples

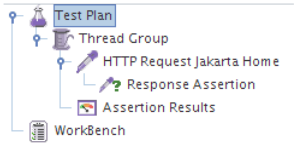


Figure 14 - Test Plan

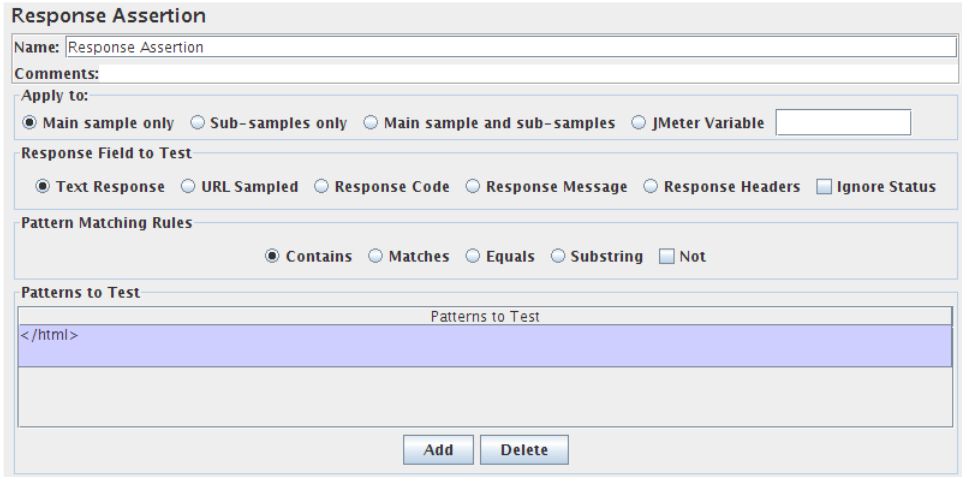


Figure 15 - Assertion Control Panel with Pattern

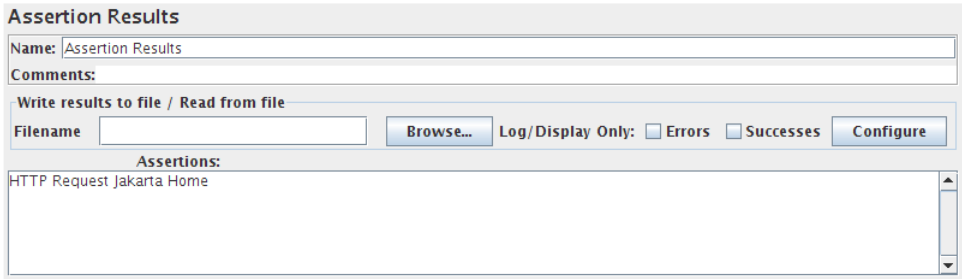


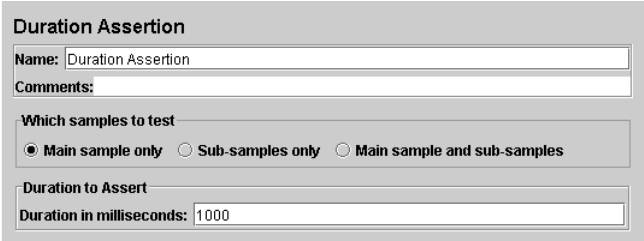
Figure 16 - Assertion Listener Results (Pass)

Figure 17 - Assertion Listener Results (Fail)

18.5.2 Duration Assertion

The Duration Assertion tests that each response was received within a given amount of time. Any response that takes longer tl of milliseconds (specified by the user) is marked as a failed response.

Control Panel



Parameters

Attribute	Description	Re
Name	Descriptive name for this element that is shown in the tree.	No
Duration in Milliseconds	The maximum number of milliseconds each response is allowed before being marked as failed.	Ye

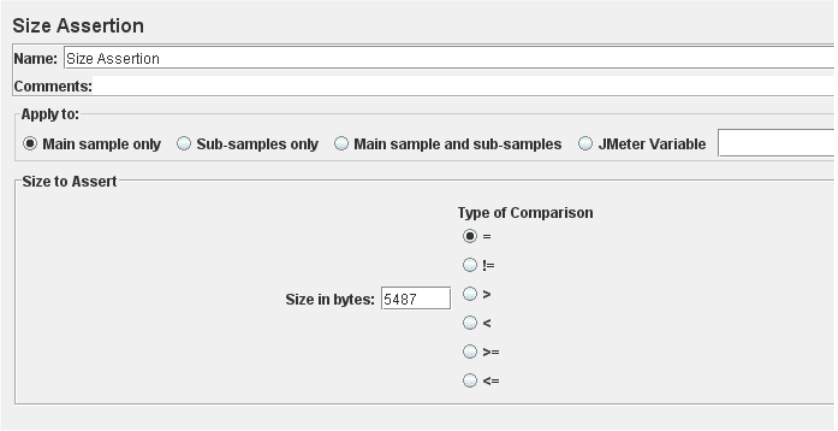
---

18.5.3 Size Assertion

The Size Assertion tests that each response contains the right number of bytes in it. You can specify that the size be equal to, g
or not equal to a given number of bytes.

Since JMeter 2.3RC3, an empty response is treated as being 0 bytes rather than reported as an error.

Control Panel



Parameters

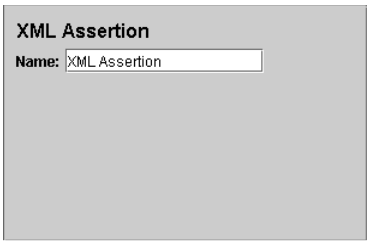
Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Apply to:	This is for use with samplers that can generate sub-samples, e.g. HTTP Sampler with embedded resources, Reader or samples generated by the Transaction Controller. • Main sample only - assertion only applies to the main sample • Sub-samples only - assertion only applies to the sub-samples • Main sample and sub-samples - assertion applies to both. • JMeter Variable - assertion is to be applied to the contents of the named variable
Size in bytes	The number of bytes to use in testing the size of the response (or value of the JMeter variable).
Type of Comparison	Whether to test that the response is equal to, greater than, less than, or not equal to, the number of bytes sp

---

18.5.4 XML Assertion

The XML Assertion tests that the response data consists of a formally correct XML document. It does not validate the XML b
schema or do any further validation.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this element that is shown in the tree.	No

---

18.5.5 BeanShell Assertion

The BeanShell Assertion allows the user to perform assertion checking using a BeanShell script.

For full details on using BeanShell, please see the BeanShell website.

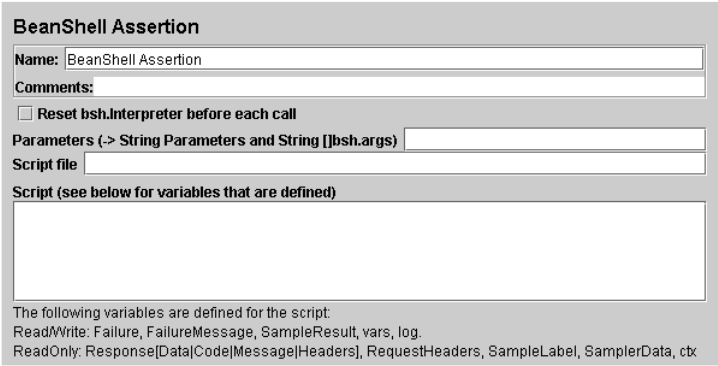
Note that a different Interpreter is used for each independent occurence of the assertion in each thread in a test script, but the s for subsequent invocations. This means that variables persist across calls to the assertion.

All Assertions are called from the same thread as the sampler.

If the property "beanshell.assertion.init" is defined, it is passed to the Interpreter as the name of a sourced file. This can be use methods and variables. There is a sample init file in the bin directory: BeanShellAssertion.bshrc

The test element supports the ThreadListener and TestListener methods. These should be defined in the initialisation file. See BeanShellListeners.bshrc for example definitions.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree. The name is stored in the script variable Label
Reset bsh.Interpreter before each call	If this option is selected, then the interpreter will be recreated for each sample. This may be necessary for some long running scripts. For further information, see Best Practices - BeanShell scripting .
Parameters	Parameters to pass to the BeanShell script. The parameters are stored in the following variables: • Parameters - string containing the parameters as a single variable • bsh.args - String array containing parameters, split on white-space
Script file	A file containing the BeanShell script to run. This overrides the script. The file name is stored in the script variable FileName
Script	The BeanShell script to run. The return value is ignored.

There's a sample script you can try.

Before invoking the script, some variables are set up in the BeanShell interpreter. These are strings unless otherwise noted:

- log - the Logger Object. (e.g.) log.warn("Message"[,Throwable])
- SampleResult - the SampleResult Object; read-write
- Response - the response Object; read-write

- Failure - boolean; read-write; used to set the Assertion status
- FailureMessage - String; read-write; used to set the Assertion message
- ResponseData - the response body (byte [])
- ResponseCode - e.g. 200
- ResponseMessage - e.g. OK
- ResponseHeaders - contains the HTTP headers
- RequestHeaders - contains the HTTP headers sent to the server
- SampleLabel
- SamplerData - data that was sent to the server
- ctx - JMeterContext
- vars - JMeterVariables - e.g. vars.get("VAR1"); vars.put("VAR2","value"); vars.putObject("OBJ1",new Object());
- props - JMeterProperties - e.g. props.get("START.HMS"); props.put("PROP1","1234");

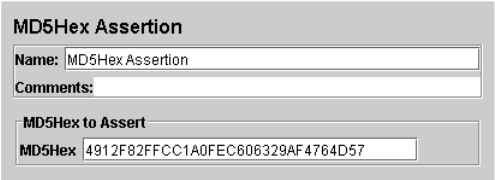
The following methods of the Response object may be useful:

- setStopThread(boolean)
- setStopTest(boolean)
- String getSampleLabel()
- setSampleLabel(String)

18.5.6 MD5Hex Assertion

The MD5Hex Assertion allows the user to check the MD5 hash of the response data.

Control Panel



Parameters

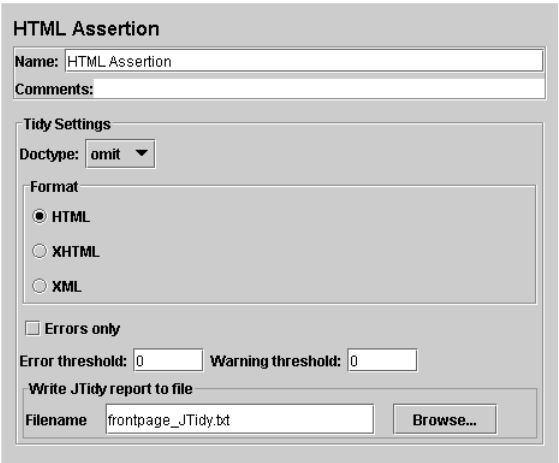
Attribute	Description	Required
Name	Descriptive name for this element that is shown in the tree.	No
MD5 sum	32 hex digits representing the MD5 hash (case not significant)	Yes

---

18.5.7 HTML Assertion

The HTML Assertion allows the user to check the HTML syntax of the response data using JTidy.

Control Panel



Parameters

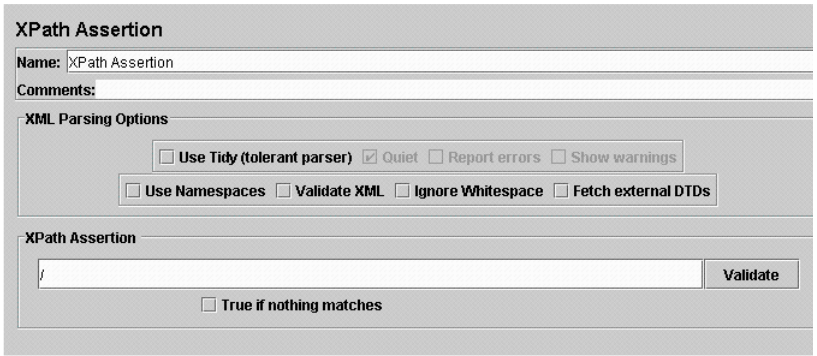
Attribute	Description	Required
Name	Descriptive name for this element that is shown in the tree.	No

doctype	omit/auto/strict/loose	Yes
Format	HTML, XHTML or XML	Yes
Errors only	Only take note of errors?	Yes
Error threshold	Number of errors allowed before classing the response as failed	Yes
Warning threshold	Number of warnings allowed before classing the response as failed	Yes
Filename	Name of file to which report is written	No

18.5.8 XPath Assertion

The XPath Assertion tests a document for well formedness, has the option of validating against a DTD, or putting the docume testing for an XPath. If that XPath exists, the Assertion is true. Using "/" will match any well-formed document, and is the def The assertion also supports boolean expressions, such as "count(//*error)=2". See http://www.w3.org/TR/xpath for more infor

Control Panel



Parameters

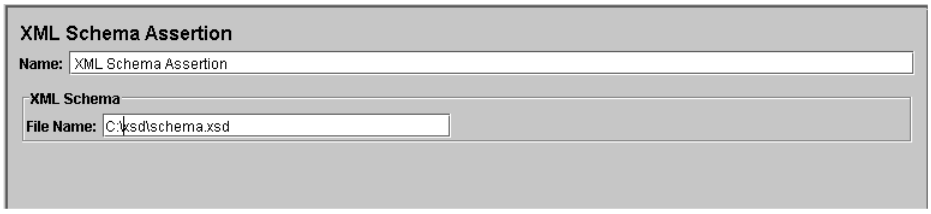
Attribute	Description	Required
Name	Descriptive name for this element that is shown in the tree.	No
Use Tidy (tolerant parser)	Use Tidy, i.e. be tolerant of XML/HTML errors	Yes
Quiet	Sets the Tidy Quiet flag	If Tidy is selected
Report Errors	If a Tidy error occurs, then set the Assertion accordingly	If Tidy is selected
Show warnings	Sets the Tidy showWarnings option	If Tidy is selected
Use Namespaces	Should namespaces be honoured?	If Tidy is not selected
Validate XML	Check the document against its schema.	If Tidy is not selected
Ignore Whitespace	Ignore Element Whitespace.	If Tidy is not selected
Fetch External DTDs	If selected, external DTDs are fetched.	If Tidy is not selected
XPath Assertion	XPath to match in the document.	Yes
True if nothing matches	True if a XPath expression is not matched	No

The non-tolerant parser can be quite slow, as it may need to download the DTD etc.

18.5.9 XML Schema Assertion

The XML Schema Assertion allows the user to validate a response against an XML Schema.

Control Panel



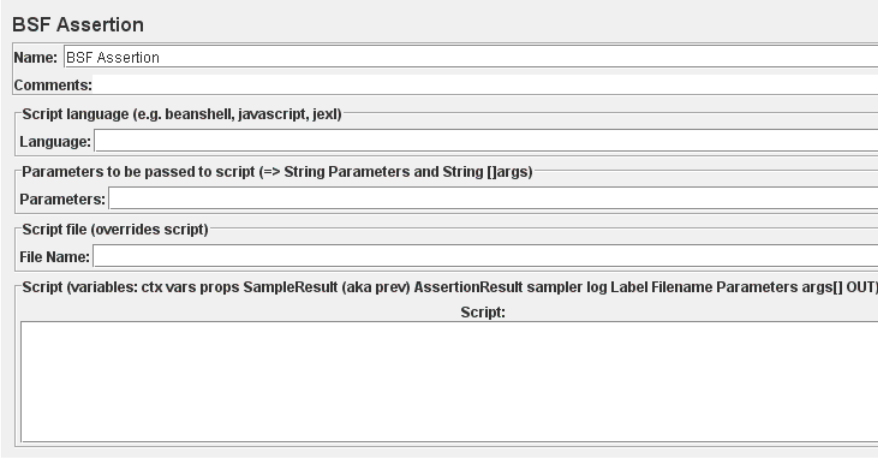
Parameters

Attribute	Description	Required
Name	Descriptive name for this element that is shown in the tree.	No
File Name	Specify XML Schema File Name	Yes

18.5.10 BSF Assertion

The BSF Assertion allows BSF script code to be used to check the status of the previous sample.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this element that is shown in the tree.	No
Language	The BSF language to be used	Yes
Parameters	Parameters to pass to the script. The parameters are stored in the following variables: • Parameters - string containing the parameters as a single variable • args - String array containing parameters, split on white-space	No
Script file	A file containing the script to run.	No
Script	The script to run.	Yes (unless script file is pr

The script (or file) is processed using the BSFEngine.exec() method, which does not return a value.

The following variables are set up for use by the script:

- log - (Logger) - can be used to write to the log file
- Label - the String Label
- Filename - the script file name (if any)
- Parameters - the parameters (as a String)
- args[] - the parameters as a String array (split on whitespace)
- ctx - (JMeterContext) - gives access to the context
- vars - (JMeterVariables) - gives read/write access to variables: vars.get(key); vars.put(key,val); vars.putObject("OBJ1", vars.getObject("OBJ2");
- props - (JMeterProperties) - e.g. props.get("START.HMS"); props.put("PROP1","1234");
- SampleResult, prev - (SampleResult) - gives access to the previous SampleResult (if any)
- sampler - (Sampler)- gives access to the current sampler
- OUT - System.out - e.g. OUT.println("message")
- AssertionResult - the assertion result

The script can check various aspects of the SampleResult. If an error is detected, the script should use AssertionResult.setFail ("message") and AssertionResult.setFailure(true).

For futher details of all the methods available on each of the above variables, please check the Javadoc

---

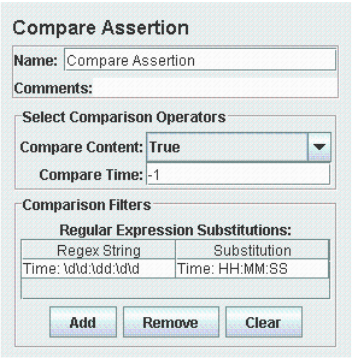
18.5.11 JSR223 Assertion

The JSR223 Assertion allows JSR223 script code to be used to check the status of the previous sample. For details, see BSF A

18.5.12 Compare Assertion

The Compare Assertion can be used to compare sample results within its scope. Either the contents or the elapsed time can be contents can be filtered before comparison. The assertion comparisons can be seen in the Comparison Assertion Visualizer .

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Compare Content	Whether or not to compare the content (response data)
Compare Time	If the value is >=0, then check if the response time difference is no greater than the value. I.e. if the value is response times must be exactly equal.
Comparison Filters	Filters can be used to remove strings from the content comparison. For example, if the page has a time-stamp be matched with: "Time: \d\d:\d\d:\d\d" and replaced with a dummy fixed time "Time: HH:MM:SS".

18.5.13 SMIME Assertion

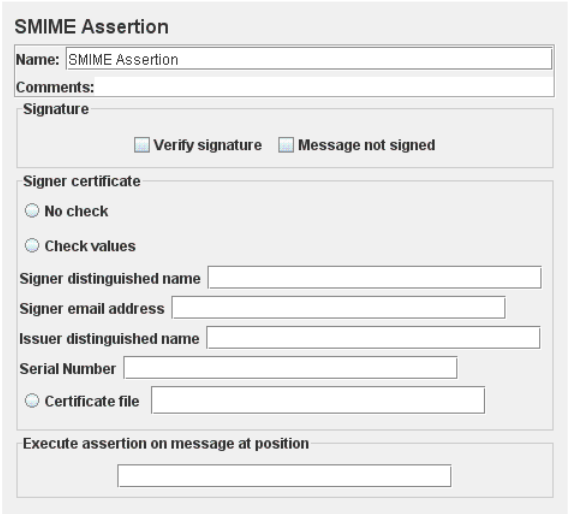
The SMIME Assertion can be used to evaluate the sample results from the Mail Reader Sampler. This assertion verifies if the message is signed or not. The signature can also be verified against a specific signer certificate. As this is a functionality that i needed by most users, additional jars need to be downloaded and added to JMETER_HOME/lib :

- bcmail-xxx.jar (BouncyCastle SMIME/CMS)
- bcprov-xxx.jar (BouncyCastle Provider)

These need to be downloaded from BouncyCastle.

If using the Mail Reader Sampler , please ensure that you select "Store the message using MIME (raw)" otherwise the Asserti process the message correctly.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Verify Signature	If selected, the asertion will verify if it is a valid signature according to the parameters defined in the Signer C box.
Message not signed	Whether or not to expect a signature in the message
Signer Cerificate	"No Check" means that it wil not perform signature verification. "Check values" is used to verify the signature inputs provided. And "Certificate file" will perform the verification against a specific certificate file.
Message Position	The Mail sampler can retrieve multiple messages in a single sample. Use this field to specify which message w checked. Messages are numbered from 0, so 0 means the first message. Negative numbers count from the LAS message; -1 means LAST, -2 means penultimate etc.

---

^

18.6 Timers

Note that timers are processed **before** each sampler in the scope in which they are found; if there are several timers in the sam will be processed **before each** sampler.

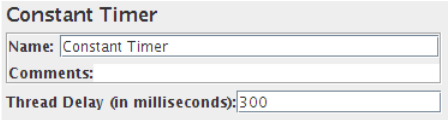
Timers are only processed in conjunction with a sampler. A timer which is not in the same scope as a sampler will not be proc

To apply a timer to a single sampler, add the timer as a child element of the sampler. The timer will be applied before the sam apply a timer after a sampler, either add it to the next sampler, or add it as the child of a Test Action Sampler.

18.6.1 Constant Timer

If you want to have each thread pause for the same amount of time between requests, use this timer.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this timer that is shown in the tree.	No
Thread Delay	Number of milliseconds to pause.	Yes

18.6.2 Gaussian Random Timer

This timer pauses each thread request for a random amount of time, with most of the time intervals ocurring near a particular is the sum of the Gaussian distributed value (with mean 0.0 and standard deviation 1.0) times the deviation value you specify,

Control Panel

Parameters

Attribute	Description	Required
Name	Descriptive name for this timer that is shown in the tree	No
Deviation	Deviation in milliseconds.	Yes

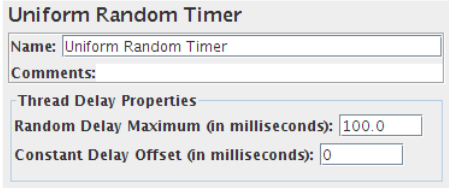
Constant Delay Offset	Number of milliseconds to pause in addition to the random delay.	Yes

---

18.6.3 Uniform Random Timer

This timer pauses each thread request for a random amount of time, with each time interval having the same probability of occ
is the sum of the random value and the offset value.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this timer that is shown in the tree.	No
Random Delay Maximum	Maxium random number of milliseconds to pause.	Yes
Constant Delay Offset	Number of milliseconds to pause in addition to the random delay.	Yes

---

18.6.4 Constant Throughput Timer

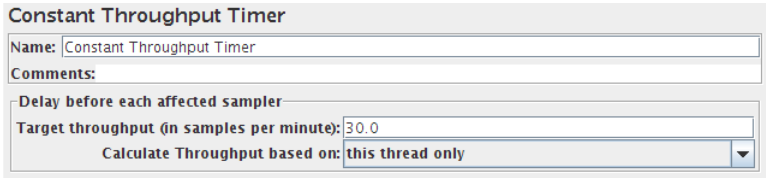
This timer introduces variable pauses, calculated to keep the total throughput (in terms of samples per minute) as close as poss
Of course the throughput will be lower if the server is not capable of handling it, or if other timers or time-consuming test eler

N.B. although the Timer is called the Constant Throughput timer, the throughput value does not need to be constant. It can be
variable or function call, and the value can be changed during a test. The value can be changed in various ways:

- using a counter variable
- using a JavaScript or BeanShell function to provide a changing value
- using the remote BeanShell server to change a JMeter property

See Best Practices for further details. Note that the throughput value should not be changed too often during a test - it will tak
value to take effect.

Control Panel



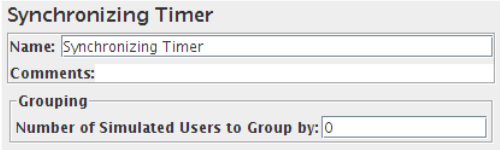
Parameters

Attribute	Description
Name	Descriptive name for this timer that is shown in the tree.
Target Throughput	Throughput we want the timer to try to generate.
Calculate Throughput based on	• this thread only - each thread will try to maintain the target throughput. The overall throughput wil proportional to the number of active threads. • all active threads in current thread group - the target throughput is divided amongst all the active th the group. Each thread will delay as needed, based on when it last ran. • all active threads - the target throughput is divided amongst all the active threads in all Thread Gro thread will delay as needed, based on when it last ran. In this case, each other Thread Group will n Constant Throughput timer with the same settings. • all active threads in current thread group (shared) - as above, but each thread is delayed based on w thread in the group last ran. • all active threads (shared) - as above; each thread is delayed based on when any thread last ran.

18.6.5 Synchronizing Timer

The purpose of the SyncTimer is to block threads until X number of threads have been blocked, and then they are all released can thus create large instant loads at various points of the test plan.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this timer that is shown in the tree.	No
Number of Simultaneous Users to Group by	Number of threads to release at once.	Yes

---

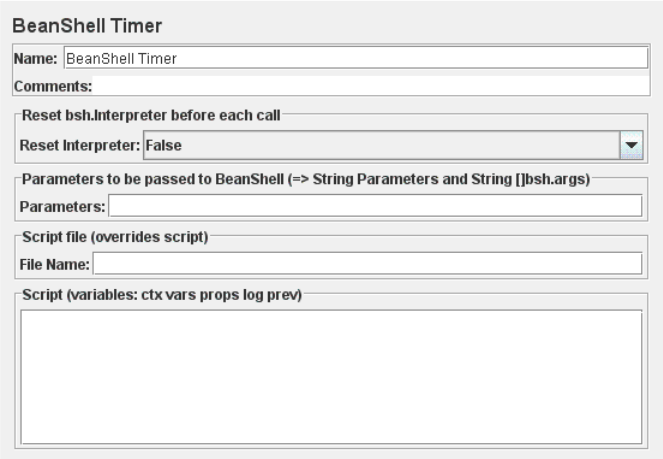
18.6.6 BeanShell Timer

The BeanShell Timer can be used to generate a delay.

For full details on using BeanShell, please see the BeanShell website.

The test element supports the ThreadListener and TestListener methods. These should be defined in the initialisation file. See BeanShellListeners.bshrc for example definitions.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree. The name is stored in the script variable Label
Reset bsh.Interpreter before each call	If this option is selected, then the interpreter will be recreated for each sample. This may be necessary for some long running scripts. For further information, see Best Practices - BeanShell scripting .
Parameters	Parameters to pass to the BeanShell script. The parameters are stored in the following variables: • Parameters - string containing the parameters as a single variable • bsh.args - String array containing parameters, split on white-space
Script file	A file containing the BeanShell script to run. The file name is stored in the script variable FileNam The return value is used as the number of milliseconds to wait.
Script	The BeanShell script. The return value is used as the number of milliseconds to wait.

Before invoking the script, some variables are set up in the BeanShell interpreter:

- log - (Logger) - can be used to write to the log file

- ctx - (JMeterContext) - gives access to the context
- vars - (JMeterVariables) - gives read/write access to variables: vars.get(key); vars.put(key,val); vars.putObject("OBJ1",
- props - (JMeterProperties) - e.g. props.get("START.HMS"); props.put("PROP1","1234");
- prev - (SampleResult) - gives access to the previous SampleResult (if any)

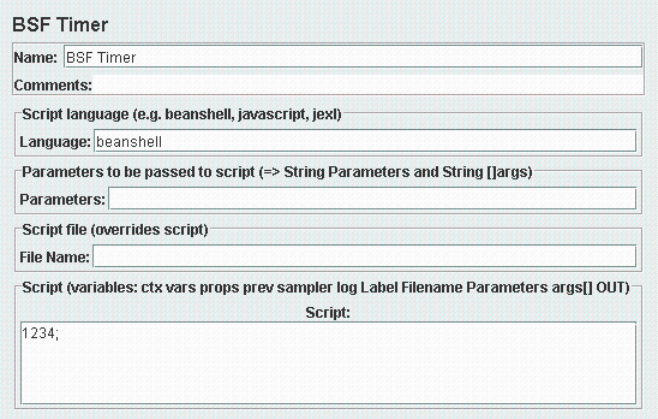
For details of all the methods available on each of the above variables, please check the Javadoc

If the property **beanshell.timer.init** is defined, this is used to load an initialisation file, which can be used to define methods e BeanShell script.

18.6.7 BSF Timer

The BSF Timer can be used to generate a delay using a BSF scripting language.

Control Panel



Parameters

Attribute	Description	
Name	Descriptive name for this element that is shown in the tree.	No
ScriptLanguage	The scripting language to be used.	Yes
Parameters	Parameters to pass to the script. The parameters are stored in the following variables: • Parameters - string containing the parameters as a single variable • args - String array containing parameters, split on white-space	No
Script file	A file containing the script to run. The return value is converted to a long integer and used as the number of milliseconds to wait.	No
Script	The script. The return value is used as the number of milliseconds to wait.	Yes (u provid

Before invoking the script, some variables are set up in the script interpreter:

- log - (Logger) - can be used to write to the log file
- ctx - (JMeterContext) - gives access to the context
- vars - (JMeterVariables) - gives read/write access to variables: vars.get(key); vars.put(key,val); vars.putObject("OBJ1",
- props - (JMeterProperties) - e.g. props.get("START.HMS"); props.put("PROP1","1234");
- sampler - the current Sampler
- Label - the name of the Timer
- Filename - the file name (if any)
- OUT - System.out

For details of all the methods available on each of the above variables, please check the Javadoc

---

18.6.8 JSR223 Timer

The JSR223 Timer can be used to generate a delay using a JSR223 scripting language, For details, see BSF Timer .

---

^

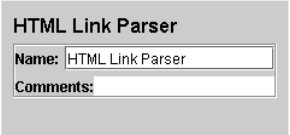
18.7 Pre Processors

Preprocessors are used to modify the Samplers in their scope.

18.7.1 HTML Link Parser

This modifier parses HTML response from the server and extracts links and forms. A URL test sample that passes through thi examined to see if it "matches" any of the links or forms extracted from the immediately previous response. It would then repl URL test sample with appropriate values from the matching link or form. Perl-type regular expressions are used to find match

Control Panel



Matches are performed using protocol, host, path and parameter names. The target sampler cannot contain parameters that are not in the response links.

Spidering Example

Consider a simple example: let's say you wanted JMeter to "spider" through your site, hitting link after link parsed from the H your server (this is not actually the most useful thing to do, but it serves as a good example). You would create a Simple Cont "HTML Link Parser" to it. Then, create an HTTP Request, and set the domain to ".*", and the path likewise. This will cause y match with any link found on the returned pages. If you wanted to restrict the spidering to a particular domain, then change th one you want. Then, only links to that domain will be followed.

Poll Example

A more useful example: given a web polling application, you might have a page with several poll options as radio buttons for say the values of the poll options are very dynamic - maybe user generated. If you wanted JMeter to test the poll, you could ei samples with hardcoded values chosen, or you could let the HTML Link Parser parse the form, and insert a random poll optio sample. To do this, follow the above example, except, when configuring your Web Test controller's URL options, be sure to c method. Put in hard-coded values for the domain, path, and any additional form parameters. Then, for the actual radio button p name (let's say it's called "poll_choice"), and then ".*" for the value of that parameter. When the modifier examines this URL that it "matches" the poll form (and it shouldn't match any other form, given that you've specified all the other aspects of the U it will replace your form parameters with the matching parameters from the form. Since the regular expression ".*" will match modifier will probably have a list of radio buttons to choose from. It will choose at random, and replace the value in your URI time through the test, a new random value will be chosen.

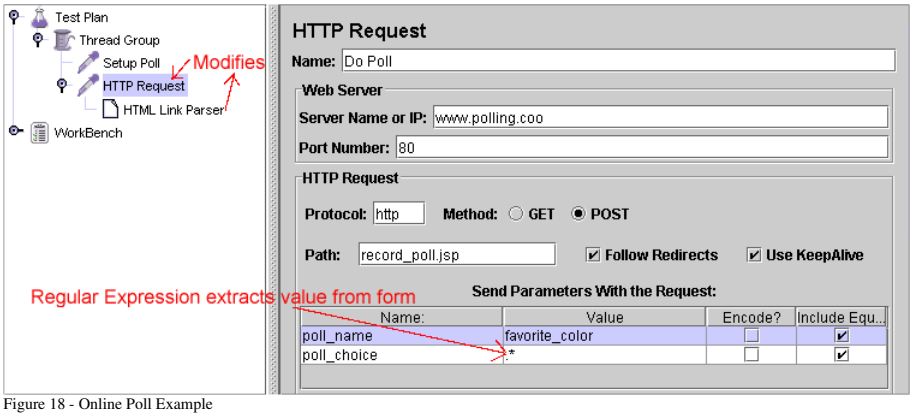


Figure 18 - Online Poll Example

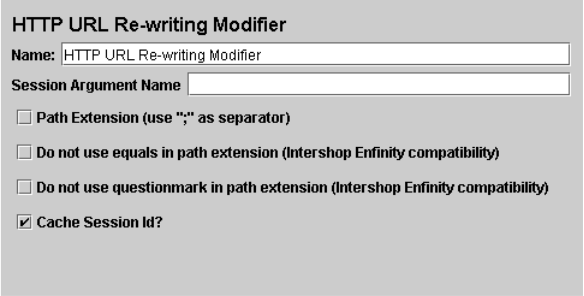
One important thing to remember is that you must create a test sample immediately prior that will return an HTML page with the links and forms that are relevant to your dynamic test sample.

18.7.2 HTTP URL Re-writing Modifier

This modifier works similarly to the HTML Link Parser, except it has a specific purpose for which it is easier to use than the I and more efficient. For web applications that use URL Re-writing to store session ids instead of cookies, this element can be a ThreadGroup level, much like the HTTP Cookie Manager . Simply give it the name of the session id parameter, and it will fin add the argument to every request of that ThreadGroup.

Alternatively, this modifier can be attached to select requests and it will modify only them. Clever users will even determine t
be used to grab values that elude the HTML Link Parser .

Control Panel



Parameters

Attribute	Description
Name	Descriptive name given to this element in the test tree.
Session Argument Name	The name of the parameter to grab from previous response. This modifier will find the paramete anywhere it exists on the page, and grab the value assigned to it, whether it's in an HREF or a fo
Path Extension	Some web apps rewrite URLs by appending a semi-colon plus the session id parameter. Check t that is so.
Do not use equals in path extension	Some web apps rewrite URLs without using an "=" sign between the parameter name and value Intershop Enfinity).
Do not use questionmark in path extension	Prevents the query string to end up in the path extension (such as Intershop Enfinity).
Cache Session Id?	Should the value of the session Id be saved for later use when the session Id is not present?

18.7.3 HTML Parameter Mask

***** This element is deprecated. Use Counter instead *****

The HTML Parameter Mask is used to generate unique values for HTML arguments. By specifying the name of the parameter
suffix, and counter parameters, this modifier will generate values of the form " `name=prefixcountersuffix` ". Any HTTP R
it will replace any parameter with the same name or add the appropriate parameter to the requests list of arguments.

The value of the argument in your HTTP Request must be a '*' in order for the HTML Parameter Mask Modifier to replace it.

As an example, the username for a login script could be modified to send a series of values such as:

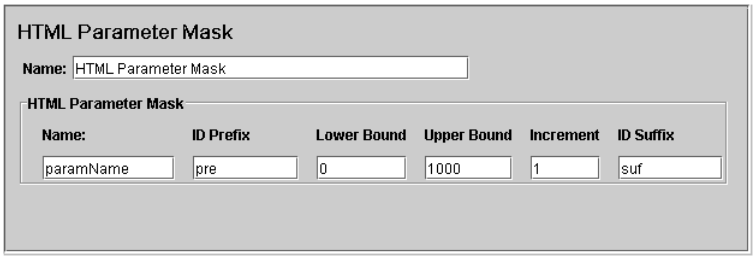
user_1

user_2

user_3

user_4, etc.

Control Panel



Parameters

Attribute	Description	Requir
Name	Descriptive name given to this element in the test tree.	No
Name (second appearing)	The name of the parameter to modify or add to the HTTP Request.	Yes
ID Prefix	A string value to prefix to every generated value.	No
Lower Bound	A number value to start the counter at.	Yes

Upper Bound	A number value to end the counter, at which point it restarts with the Lower Bound value.	Yes
Increment	Value to increment the counter by each time through.	Yes
ID Suffix	A string value to add as suffix to every generated vaue.	No

---

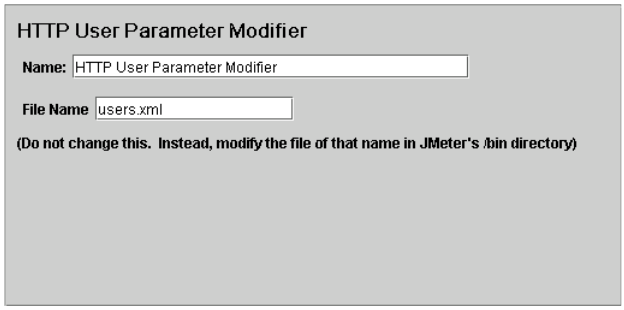
18.7.4 HTTP User Parameter Modifier

***** This element is deprecated. Use User Parameters instead *****

See also the CSV Data Set Config element, which is more suitable for large numbers of parameters

The User Parameter Modifier uses an XML file get values for HTTP arguments. Any HTTP Request that this modifier modifi the existence of the specified arguments. If found, the values for those arguments will be replaced by the values found in the x can have multiple sets of the same values. This modifier will iterate through these values in a round-robin style, thus each requ set of values until the last set of values is reached, at which point it will begin again at the first set.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name given to this element in the test tree.	No
File Name	Name of the XML file in JMeter's /bin directory that holds the value sets.	Yes

---

18.7.5 User Parameters

Allows the user to specify values for User Variables specific to individual threads.

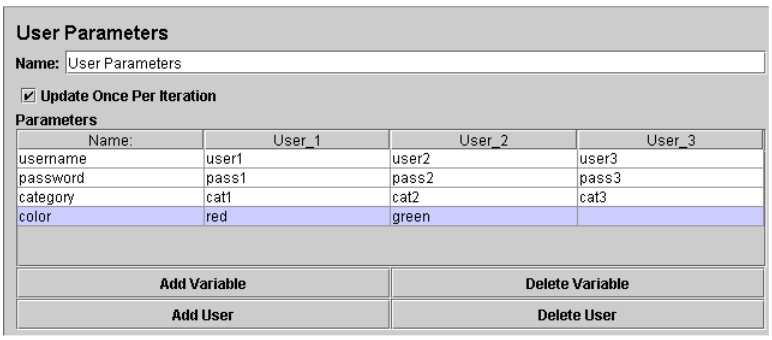
User Variables can also be specified in the Test Plan but not specific to individual threads. This panel allows you to specify a User Variable. For each thread, the variable will be assigned one of the values from the series in sequence. If there are more th values get re-used. For example, this can be used to assign a distinct user id to be used by each thread. User variables can be r of any jMeter Component.

The variable is specified by clicking the Add Variable button in the bottom of the panel and filling in the Variable name in the add a new value to the series, click the 'Add User' button and fill in the desired value in the newly added column.

Values can be accessed in any test component in the same thread group, using the function syntax : ${variable}.

See also the CSV Data Set Config element, which is more suitable for large numbers of parameters

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Update Once Per Iteration	A flag to indicate whether the User Paramters element should update its variables only once per iteration. if yo functions into the UP, then you may need greater control over how often the values of the variables are update this box checked to ensure the values are updated each time through the UP's parent controller. Uncheck the b UP will update the parameters for every sample request made within its scope .

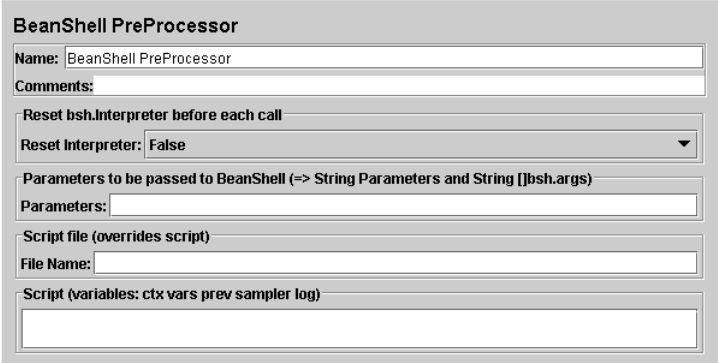
18.7.7 BeanShell PreProcessor

The BeanShell PreProcessor allows arbitrary code to be applied before taking a sample.

For full details on using BeanShell, please see the BeanShell website.

The test element supports the ThreadListener and TestListener methods. These should be defined in the initialisation file. See BeanShellListeners.bshrc for example definitions.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree. The name is stored in the script variable Label
Reset bsh.Interpreter before each call	If this option is selected, then the interpreter will be recreated for each sample. This may be necessary for some long running scripts. For further information, see Best Practices - BeanShell scripting .
Parameters	Parameters to pass to the BeanShell script. The parameters are stored in the following variables: • Parameters - string containing the parameters as a single variable • bsh.args - String array containing parameters, split on white-space
Script file	A file containing the BeanShell script to run. The file name is stored in the script variable FileName
Script	The BeanShell script. The return value is ignored.

Before invoking the script, some variables are set up in the BeanShell interpreter:

- log - (Logger) - can be used to write to the log file
- ctx - (JMeterContext) - gives access to the context
- vars - (JMeterVariables) - gives read/write access to variables: vars.get(key); vars.put(key,val); vars.putObject("OBJ1",
- props - (JMeterProperties) - e.g. props.get("START.HMS"); props.put("PROP1","1234");
- prev - (SampleResult) - gives access to the previous SampleResult (if any)
- sampler - (Sampler)- gives access to the current sampler

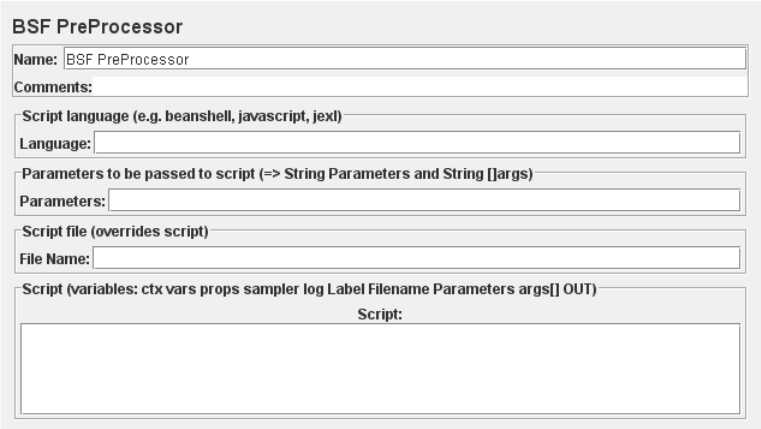
For details of all the methods available on each of the above variables, please check the Javadoc

If the property **beanshell.preprocessor.init** is defined, this is used to load an initialisation file, which can be used to define m the BeanShell script.

18.7.8 BSF PreProcessor

The BSF PreProcessor allows BSF script code to be applied before taking a sample.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this element that is shown in the tree.	No
Language	The BSF language to be used	Yes
Parameters	Parameters to pass to the script. The parameters are stored in the following variables: • Parameters - string containing the parameters as a single variable • args - String array containing parameters, split on white-space	No
Script file	A file containing the script to run.	No
Script	The script to run.	Yes (unless script file is pr

The script (or file) is processed using the BSFEngine.exec() method, which does not return a value.

The following BSF variables are set up for use by the script:

- log - (Logger) - can be used to write to the log file
- Label - the String Label
- Filename - the script file name (if any)
- Parameters - the parameters (as a String)
- args[] - the parameters as a String array (split on whitespace)
- ctx - (JMeterContext) - gives access to the context
- vars - (JMeterVariables) - gives read/write access to variables: vars.get(key); vars.put(key,val); vars.putObject("OBJ1", vars.getObject("OBJ2");
- props - (JMeterProperties) - e.g. props.get("START.HMS"); props.put("PROP1","1234");
- sampler - (Sampler)- gives access to the current sampler
- OUT - System.out - e.g. OUT.println("message")

For details of all the methods available on each of the above variables, please check the Javadoc

---

18.7.8 JSR223 PreProcessor

The JSR223 PreProcessor allows JSR223 script code to be applied before taking a sample. For details, see BSF PreProcessor .

---

^

18.8 Post-Processors

As the name suggests, Post-Processors are applied after samplers. Note that they are applied to **all** the samplers in the same sc post-processor is applied only to a particular sampler, add it as a child of the sampler.

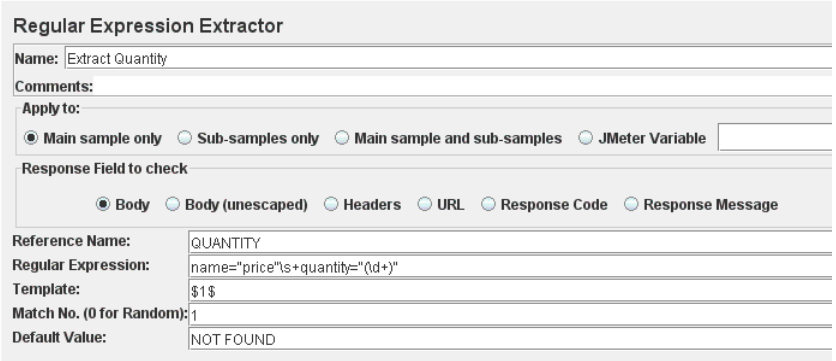
Note: Unless documented otherwise, Post-Processors are not applied to sub-samples (child samples) - only to the parent sampl and BeanShell post-processors, the script can retrieve sub-samples using the method `prev.getSubResults()` which returns a SampleResults. The array will be empty if there are none.

Post-Processors are run before Assertions, so they do not have access to any Assertion Results, nor will the sample status refle Assertions. If you require access to Assertion Results, try using a Listener instead. Also note that the variable JMeterThread.la "true" or "false" after all Assertions have been run.

18.8.1 Regular Expression Extractor

Allows the user to extract values from a server response using a Perl-type regular expression. As a post-processor, this elemen each Sample request in its scope, applying the regular expression, extracting the requested values, generate the template string into the given variable name.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Apply to:	This is for use with samplers that can generate sub-samples, e.g. HTTP Sampler with embedded resources, Ma Reader or samples generated by the Transaction Controller. • Main sample only - only applies to the main sample • Sub-samples only - only applies to the sub-samples • Main sample and sub-samples - applies to both. • JMeter Variable - assertion is to be applied to the contents of the named variable Matching is applied to all qualifying samples in turn. For example if there is a main sample and 3 sub-samples, of which contains a single match for the regex, (i.e. 4 matches in total). For match number = 3, Sub-samples or the extractor will match the 3rd sub-sample. For match number = 3, Main sample and sub-samples, the extracto match the 2nd sub-sample (1st match is main sample). For match number = 0 or negative, all qualifying sampl will be processed. For match number > 0, matching will stop as soon as enough matches have been found.
Response Field to check	The following response fields can be checked: • Body - the body of the response, e.g. the content of a web-page (excluding headers) • Body (unescaped) - the body of the response, with all Html escape codes replaced. Note that Html escap processed without regard to context, so some incorrect substitutions may be made. • Headers - may not be present for non-HTTP samples • URL • Response Code - e.g. 200 • Response Message - e.g. OK Headers can be useful for HTTP samples; it may not be present for other sample types.
Reference Name	The name of the JMeter variable in which to store the result. Also note that each group is stored as [refname]_g where [refname] is the string you entered as the reference name, and # is the group number, where group 0 is th entire match, group 1 is the match from the first set of parentheses, etc.
Regular Expression	The regular expression used to parse the response data. This must contain at least one set of parentheses "()" to capture a portion of the string, unless using the group 0. Do not enclose the expression in / / - unless of cours want to match these characters as well.
Template	The template used to create a string from the matches found. This is an arbitrary string with special elements to to groups within the regular expression. The syntax to refer to a group is: '1' to refer to group 1, '2' to refer group 2, etc. 0 refers to whatever the entire expression matches.
Match No.	Indicates which match to use. The regular expression may match multiple times. • Use a value of zero to indicate JMeter should choose a match at random. • A positive number N means to select the nth match. • Negative numbers are used in conjunction with the ForEach controller - see below.
Default Value	If the regular expression does not match, then the reference variable will be set to the default value. This is particularly useful for debugging tests. If no default is provided, then it is difficult to tell whether the regular expression did not match, or the RE element was not processed or maybe the wrong variable is being used. However, if you have several test elements that set the same variable, you may wish to leave the variable uncha if the expression does not match. In this case, remove the default value once debugging is complete.

If the match number is set to a non-negative number, and a match occurs, the variables are set as follows:

- refName - the value of the template
- refName_gn, where n=0,1,2 - the groups for the match

- refName_g - the number of groups in the Regex (excluding 0)

If no match occurs, then the refName variable is set to the default (unless this is absent). Also, the following variables are rem

- refName_g0
- refName_g1
- refName_g

If the match number is set to a negative number, then all the possible matches in the sampler data are processed. The variables

- refName_matchNr - the number of matches found; could be 0
- refName_n, where n = 1,2,3 etc - the strings as generated by the template
- refName_n_gm, where m=0,1,2 - the groups for match n
- refName - always set to the default value
- refName_gn - not set

Note that the refName variable is always set to the default value in this case, and the associated group variables are not set.

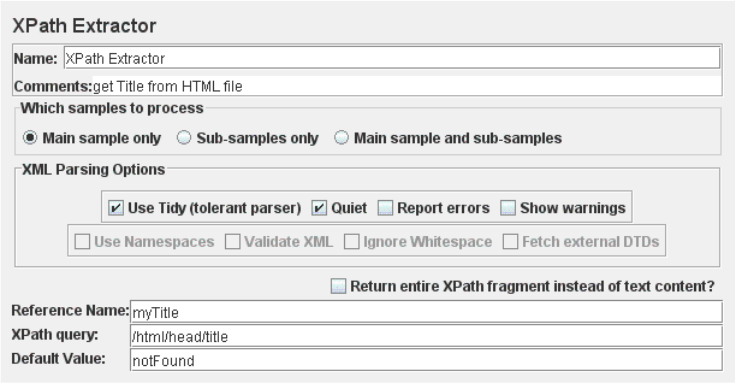
See also Response Assertion for some examples of how to specify modifiers, and for further information on JMeter regular ex

---

18.8.2 XPath Extractor

This test element allows the user to extract value(s) from structured response - XML or (X)HTML - using XPath query langua

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Apply to:	This is for use with samplers that can generate sub-samples, e.g. HTTP Sampler with embedded resources, Mail Reader or samples generated by the Transaction Controller. • Main sample only - only applies to the main sample • Sub-samples only - only applies to the sub-samples • Main sample and sub-samples - applies to both. XPath matching is applied to all qualifying samples in turn, and all the matching results will be ret
Use Tidy (tolerant parser)	If checked use Tidy to parse HTML response into XHTML. • "Use Tidy" should be checked on for HTML response. Such response is converted to valid XHTML (XML compatible HTML) using Tidy • "Use Tidy" should be unchecked for both XHTML or XML response (for example RSS)
Quiet	Sets the Tidy Quiet flag
Report Errors	If a Tidy error occurs, then set the Assertion accordingly
Show warnings	Sets the Tidy showWarnings option
Use Namespaces	If checked, then the XML parser will use namespace resolution. Note that currently only namespac declared on the root element will be recognised. A later version of JMeter may support user-defini additional workspace names. Meanwhile, a work-round is to replace: //mynamespace:tagname by //*[local-name()='tagname' and namespace-uri()='uri-for-namespace']

	where "uri-for-namespace" is the uri for the "mynamespace" namespace. (not applicable if Tidy is selected)
Validate XML	Check the document against its schema.
Ignore Whitespace	Ignore Element Whitespace.
Fetch External DTDs	If selected, external DTDs are fetched.
Return entire XPath fragment instead of text content?	If selected, the fragment will be returned rather than the text content. For example //title would return "<title>Apache JMeter</title>" rather than "Apache JMeter". In this case, //title/text() would return "Apache JMeter".
Reference Name	The name of the JMeter variable in which to store the result.
XPath Query	Element query in XPath language. Can return more than one match.
Default Value	Default value returned when no match found. It is also returned if the node has no value and the fra option is not selected.

To allow for use in a ForEach Controller, the following variables are set on return:

- refName - set to first (or only) match; if no match, then set to default
- refName_matchNr - set to number of matches (may be 0)
- refName_n - n=1,2,3 etc. Set to the 1st, 2nd 3rd match etc.

Note: The next refName_n variable is set to null - e.g. if there are 2 matches, then refName_3 is set to null, and if there are no refName_1 is set to null.

XPath is query language targeted primarily for XSLT transformations. However it is usefull as generic query language for stru XPath Reference or XPath specification for more information. Here are few examples:

/html/head/title
    extracts title element from HTML response
/book/page[2]
    extracts 2nd page from a book
/book/page
    extracts all pages from a book
//form[@name='countryForm']//select[@name='country']/option[text()='Czech Republic'])/@value
    extracts value attribute of option element that match text 'Czech Republic' inside of select element with name attribute '
    with name attribute 'countryForm'

When "Use Tidy" is checked on - resulting XML document may slightly differ from original HTML response:

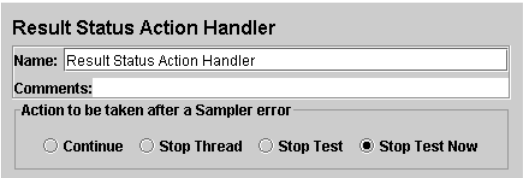
- All elements and attribute names are converted to lowercase
- Tidy attempts to correct improperly nested elements. For example - original (incorrect) `ul/font/li` becomes correct `ul/li/font`

See Tidy homepage for more information.

18.8.3 Result Status Action Handler

This test element allows the user to stop the thread or the whole test if the relevant sampler failed.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Action to be taken after a Sampler error	Determines what happens if a sampler error occurs, either because the sample itself failed or assertion failed. The possible choices are: • Continue - ignore the error and continue with the test • Stop Thread - current thread exits • Stop Test - the entire test is stopped at the end of any current samples. • Stop Test Now - the entire test is stopped abruptly. Any current samplers are interrupte possible.

18.8.4 BeanShell PostProcessor

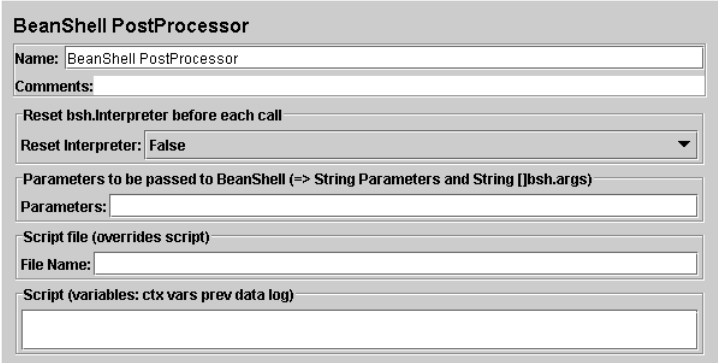
The BeanShell PreProcessor allows arbitrary code to be applied after taking a sample.

For JMeter versions after 2.2 the BeanShell Post-Processor no longer ignores samples with zero-length result data

For full details on using BeanShell, please see the BeanShell website.

The test element supports the ThreadListener and TestListener methods. These should be defined in the initialisation file. See BeanShellListeners.bshrc for example definitions.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree. The name is stored in the script variable Label
Reset bsh.Interpreter before each call	If this option is selected, then the interpreter will be recreated for each sample. This may be necessary for some long running scripts. For further information, see Best Practices - BeanShell scripting .
Parameters	Parameters to pass to the BeanShell script. The parameters are stored in the following variables: • Parameters - string containing the parameters as a single variable • bsh.args - String array containing parameters, split on white-space
Script file	A file containing the BeanShell script to run. The file name is stored in the script variable FileName
Script	The BeanShell script. The return value is ignored.

The following BeanShell variables are set up for use by the script:

- log - (Logger) - can be used to write to the log file
- ctx - (JMeterContext) - gives access to the context
- vars - (JMeterVariables) - gives read/write access to variables: vars.get(key); vars.put(key,val); vars.putObject("OBJ1",
- props - (JMeterProperties) - e.g. props.get("START.HMS"); props.put("PROP1","1234");
- prev - (SampleResult) - gives access to the previous SampleResult
- data - (byte [])- gives access to the current sample data

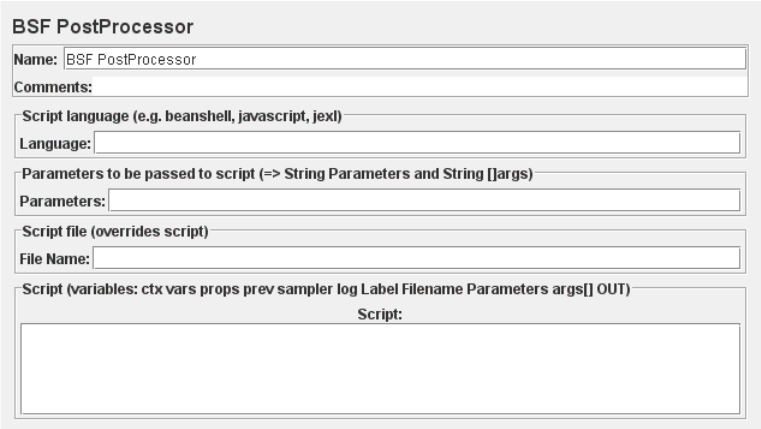
For details of all the methods available on each of the above variables, please check the Javadoc

If the property **beanshell.postprocessor.init** is defined, this is used to load an initialisation file, which can be used to define n the BeanShell script.

18.8.5 BSF PostProcessor

The BSF PostProcessor allows BSF script code to be applied after taking a sample.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this element that is shown in the tree.	No
Language	The BSF language to be used	Yes
Parameters	Parameters to pass to the script. The parameters are stored in the following variables: • Parameters - string containing the parameters as a single variable • args - String array containing parameters, split on white-space	No
Script file	A file containing the script to run.	No
Script	The script to run.	Yes (unless script file is pr

The script (or file) is processed using the BSFEngine.exec() method, which does not return a value.

Before invoking the script, some variables are set up. Note that these are BSF variables - i.e. they can be used directly in the s

- log - (Logger) - can be used to write to the log file
- Label - the String Label
- Filename - the script file name (if any)
- Parameters - the parameters (as a String)
- args[] - the parameters as a String array (split on whitespace)
- ctx - (JMeterContext) - gives access to the context
- vars - (JMeterVariables) - gives read/write access to variables: vars.get(key); vars.put(key,val); vars.putObject("OBJ1", vars.getObject("OBJ2");
- props - (JMeterProperties) - e.g. props.get("START.HMS"); props.put("PROP1","1234");
- prev - (SampleResult) - gives access to the previous SampleResult (if any)
- sampler - (Sampler)- gives access to the current sampler
- OUT - System.out - e.g. OUT.println("message")

For details of all the methods available on each of the above variables, please check the Javadoc

---

18.8.6 JSR223 PostProcessor

The JSR223 PostProcessor allows JSR223 script code to be applied after taking a sample. For details, see the BSF PostProces

---

^

18.9 Miscellaneous Features

18.9.1 Test Plan

The Test Plan is where the overall settings for a test are specified.

Static variables can be defined for values that are repeated throughout a test, such as server names. For example the variable S defined as www.example.com, and the rest of the test plan could refer to it as ${SERVER}. This simplifies changing the nam

If the same variable name is reused on one of more User Defined Variables Configuration elements, the value is set to the last plan (reading from top to bottom). Such variables should be used for items that may change between test runs, but which rem test run.

Note that the Test Plan cannot refer to variables it defines. If you need to construct other variables from the Test Plan vari
Defined Variables test element.

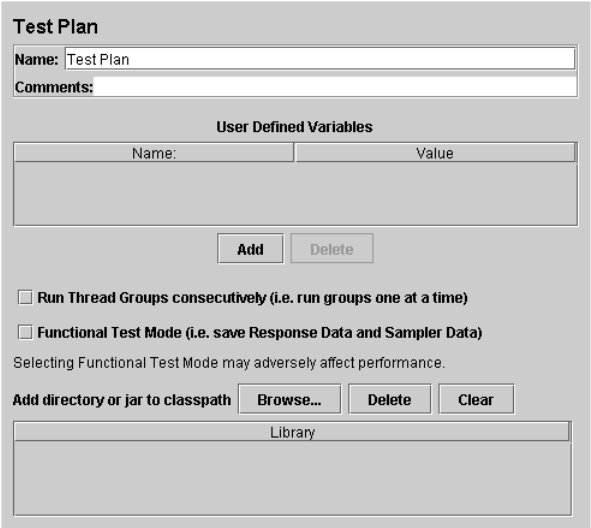
Selecting Functional Testing instructs JMeter to save the additional sample information - Response Data and Sampler Data - t
increases the resources needed to run a test, and may adversely impact JMeter performance. If more data is required for a part
then add a Listener to it, and configure the fields as required. [The option does not affect CSV result files, which cannot curre
information.]

Also, an option exists here to instruct JMeter to run the Thread Group serially rather than in parallel.

Test plan now provides an easy way to add classpath setting to a specific test plan. The feature is additive, meaning that you c
directories, but removing an entry requires restarting JMeter. Note that this cannot be used to add JMeter GUI plugins, becaus
earlier. However it can be useful for utility jars such as JDBC drivers.

JMeter properties also provides an entry for loading additional classpaths. In jmeter.properties, edit "user.classpath" to include

Control Panel



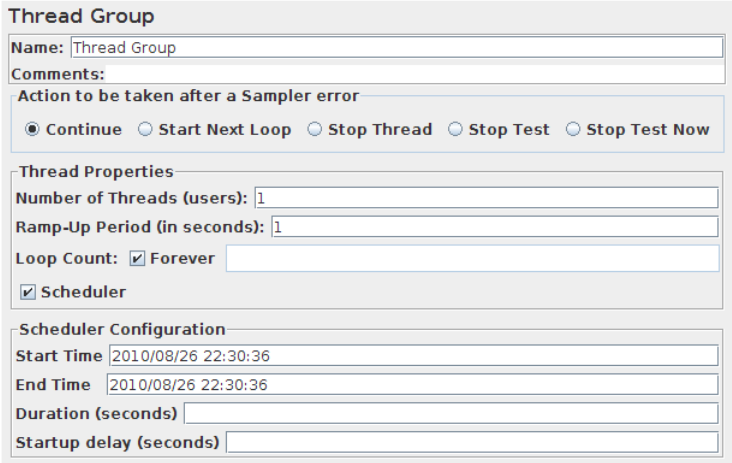
18.9.2 Thread Group

A Thread Group defines a pool of users that will execute a particular test case against your server. In the Thread Group GUI, y
number of users simulated (num of threads), the ramp up time (how long it takes to start all the threads), the number of times t
optionally, a start and stop time for the test.

See also tearDown Thread Group and setUp Thread Group .

When using the scheduler, JMeter runs the thread group until either the number of loops is reached or the duration/end-time is
occurs first. Note that the condition is only checked between samples; when the end condition is reached, that thread will stop
interrupt samplers which are waiting for a response, so the end time may be delayed arbitrarily.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this element that is shown in the tree.
Action to be taken after a Sampler error	Determines what happens if a sampler error occurs, either because the sample itself failed or an assertic failed. The possible choices are: • Continue - ignore the error and continue with the test • Start Next Loop - ignore the error, start next loop and continue with the test • Stop Thread - current thread exits • Stop Test - the entire test is stopped at the end of any current samples. • Stop Test Now - the entire test is stopped abruptly. Any current samplers are interrupted if possib
Number of Threads	Number of users to simulate.
Ramp-up Period	How long JMeter should take to get all the threads started. If there are 10 threads and a ramp-up time o 100 seconds, then each thread will begin 10 seconds after the previous thread started, for a total time of 100 seconds to get the test fully up to speed.
Loop Count	Number of times to perform the test case. Alternatively, "forever" can be selected causing the test to ru until manually stopped.
Start Time	If the scheduler checkbox is selected, one can choose an absolute start time. When you start your test, JMeter will wait until the specified start time to begin testing. Note: the Startup Delay field over-rides t - see below.
End Time	If the scheduler checkbox is selected, one can choose an absolute end time. When you start your test, JMeter will wait until the specified start time to begin testing, and it will stop at the specified end time. Note: the Duration field over-rides this - see below.
Duration (seconds)	If the scheduler checkbox is selected, one can choose a relative end time. JMeter will use this to calcula the End Time, and ignore the End Time value.
Startup delay (seconds)	If the scheduler checkbox is selected, one can choose a relative startup delay. JMeter will use this to calculate the Start Time, and ignore the Start Time value.

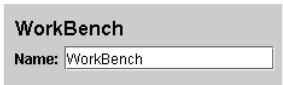
18.9.3 WorkBench

The WorkBench simply provides a place to temporarily store test elements while not in use, for copy/paste purposes, or any o desire. When you save your test plan, WorkBench items are not saved with it. Your WorkBench can be saved independently, on WorkBench and choose Save).

Certain test elements are only available on the WorkBench:

- HTTP Proxy Server
- HTTP Mirror Server
- Property Display

Control Panel



18.9.4 SSL Manager

The SSL Manager is a way to select a client certificate so that you can test applications that use Public Key Infrastructure (PK you have not set up the appropriate System properties.

Choosing a Client Certificate

You may either use a Java Key Store (JKS) format key store, or a Public Key Certificate Standard #12 (PKCS12) file for your There is a feature of the JSSE libraries that require you to have at least a six character password on your key (at least for the k comes with your JDK).

To select the client certificate, choose Options->SSL Manager from the menu bar. You will be presented with a file finder tha files by default. Your PKCS12 file must have the extension '.p12' for SSL Manager to recognize it as a PKCS12 file. Any othe like an average JKS key store. If JSSE is correctly installed, you will be prompted for the password. The text box does not hid type at this point--so make sure no one is looking over your shoulder. The current implementation assumes that the password the password for the private key of the client you want to authenticate as.

Or you can set the appropriate System properties - see the system.properties file.

The next time you run your test, the SSL Manager will examine your key store to see if it has at least one key available to it. I SSL Manager will select it for you. If there is more than one key, it currently selects the first key. There is currently no way to the keystore, so the desired key must be the first.

Things to Look Out For

You must have your Certificate Authority (CA) certificate installed properly if it is not signed by one of the five CA certificat
JDK. One method to install it is to import your CA certificate into a JKS file, and name the JKS file "jssecacerts". Place the fi
lib/security folder. This file will be read before the "cacerts" file in the same directory. Keep in mind that as long as the "jssec
certificates installed in "cacerts" will not be used. This may cause problems for you. If you don't mind importing your CA cert
"cacerts" file, then you can authenticate against all of the CA certificates installed.

18.9.5 HTTP Proxy Server

The Proxy Server allows JMeter to watch and record your actions while you browse your web application with your normal b
create test sample objects and store them directly into your test plan as you go (so you can view samples interactively while y

To use the proxy server, *add* the HTTP Proxy Server element to the workbench. Select the WorkBench element in the tree, an
element to get the Add menu (Add --> Non-Test Elements --> HTTP Proxy Server).

You also need to set up your browser to use the JMeter proxy port as the proxy for HTTP and HTTPS requests. Do not use JM
any other request types - FTP, etc. - as the JMeter proxy cannot handle them.

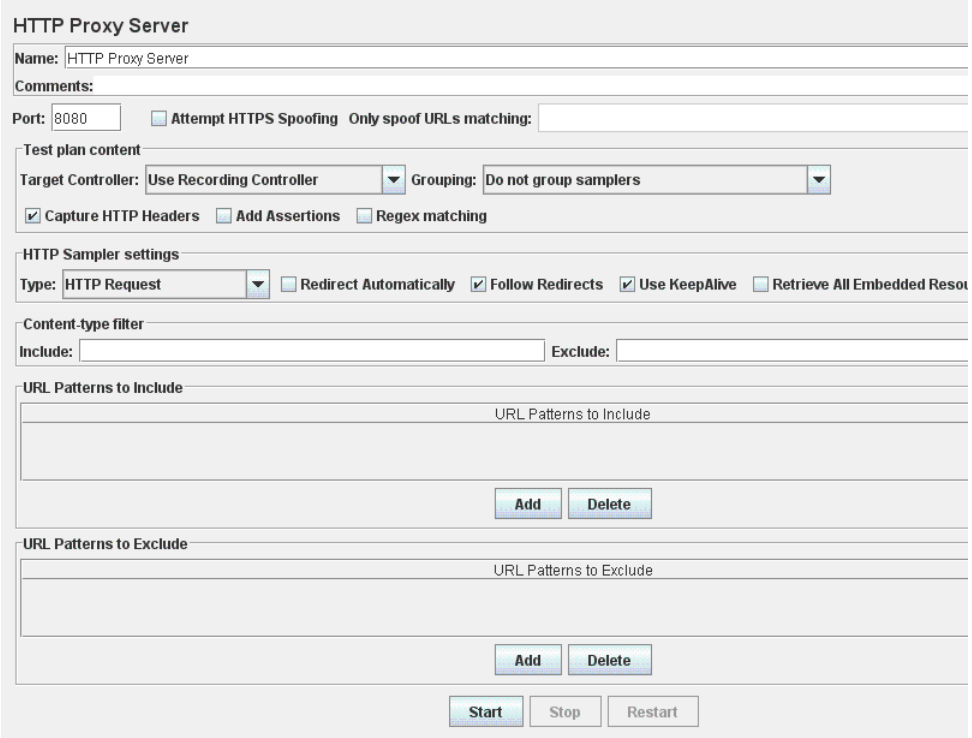
When recording HTTPS, the JMeter proxy server uses a dummy certificate to enable it to accept the SSL connection from the
certificate is not one of the certificates that browsers normally trust, and will not be for the correct host, so the browser should
asking if you want to accept the certificate or not. For example: `1) The server's name "www.example.com" does not ma
certificate's name "JMeter Proxy". Somebody may be trying to eavesdrop on you. 2) The certificate for
signed by the unknown Certificate Authority "JMeter Proxy". It is not possible to verify that this is
certificate.` You will need to accept the certificate in order to allow the JMeter Proxy to intercept the SSL traffic in order
should only accept the certificate temporarily.

The following properties can be used to change the certificate that is used:

- proxy.cert.directory - the directory in which to find the certificate (default = JMeter bin/)
- proxy.cert.file - name of the keystore file (default "proxyserver.jks")
- proxy.cert.keystorepass - keystore password (default "password")
- proxy.cert.keypassword - certificate key password (default "password")
- proxy.cert.type - the certificate type (default "JKS")
- proxy.cert.factory - the factory (default "SunX509")
- proxy.ssl.protocol - the protocol to be used (default "SSLv3")

If your browser currently uses a proxy (e.g. a company intranet may route all external requests via a proxy), then you need to tell JMeter to use that proxy before starting JMeter, using the command-line options -H and -P. This setting will also be needed when running the generated test plan.

Control Panel



Parameters

Attribute	Description
Name	Descriptive name for this controller that is shown in the tree.
Port	The port that the Proxy Server listens to. 8080 is the default, but you can change it.
Attempt HTTPS Spoofing	[Note: HTTPS spoofing should no longer be required] When you enable HTTPS spoofing, the following h • All matching (see below) http requests from the client are turned into https (between the proxy and t server). • All text response data is scanned and any occurrence of the string "https://" is replaced with "http://" default HTTPS port (443) is also removed if present. So if you want to use this feature, while you are browsing in your client, instead of typing "https://..." into t browser, type "http://...". JMeter will request and record *everything that matches* as https, whether it shoulc
Optional URL match string	If this is specified, it must be a regular expression (java.util.regex) which matches the HTTP URL(s) to be For example, if you want to spoof http://a.b.c/service/ but not http://a.b.c/images, then you could use the ex "http://a.b.c/service/.*". Note that the expression ends in ".*" because it must match the whole URL.
Target Controller	The controller where the proxy will store the generated samples. By default, it will look for a Recording C and store them there wherever it is.
Grouping	Whether to group samplers for requests from a single "click" (requests received without significant time se and how to represent that grouping in the recording: • Do not group samplers: store all recorded samplers sequentially, without any grouping. • Add separators between groups: add a controller named "----------" to create a visual separation b groups. Otherwise the samplers are all stored sequentially. • Put each group in a new controller: create a new Simple Controller for each group, and store all sam that group in it. • Store 1st sampler of each group only: only the first request in each group will be recorded. The "Foll Redirects" and "Retrieve All Embedded Resources..." flags will be turned on in those samplers. • Put each group in a new transaction controller: create a new Transaction Controller for each group, a samplers for that group in it. The property **proxy.pause** determines the minimum gap that JMeter needs between requests to treat them "clicks". The default is 1000 (milliseconds) i.e. 1 second. If you are using grouping, please ensure that you required gap between clicks.
Capture HTTP Headers	Should headers be added to the plan? If specified, a Header Manager will be added to each HTTP Sampler Proxy server always removes Cookie and Authorization headers from the generated Header Managers. By also removes If-Modified-Since and If-None-Match headers. These are used to determine if the browser ca are up to date; when recording one normally wants to download all the content. To change which additiona are removed, define the JMeter property **proxy.headers.remove** as a comma-separated list of headers.
Add Assertions	Add a blank assertion to each sampler?
Regex Matching	Use Regex Matching when replacing variables?
Type	Which type of sampler to generate (the Java default or HTTPClient)
Redirect Automatically	Set Redirect Automatically in the generated samplers?
Follow Redirects	Set Follow Redirects in the generated samplers?
Use Keep-Alive	Set Use Keep-Alive in the generated samplers?
Retrieve all Embedded Resources	Set Retrieve all Embedded Resources in the generated samplers?
Content Type filter	Filter the requests based on the content-type - e.g. "text/html [;charset=utf-8]". The fields are regular expr which are checked to see if they are contained in the content-type. [Does not have to match the entire field include filter is checked first, then the exclude filter. Samples which are filtered out will not be stored.
Patterns to Include	Regular expressions that are matched against the full URL that is sampled. Allows filtering of requests tha recorded. All requests pass through, but only those that meet the requirements of the Include/Exclude field *recorded* . If both Include and Exclude are left empty, then everything is recorded (which can result in doz samples recorded for each page, as images, stylesheets, etc are recorded). **If there is at least one entry in Include field, then only requests that match one or more Include patterns are recorded** .
Patterns to Exclude	Regular expressions that are matched against the URL that is sampled. **Any requests that match one or n Exclude pattern are *not* recorded** .
Start Button	Start the proxy server. JMeter writes the following message to the console once the proxy server has starte ready to take requests: "Proxy up and running!".
Stop Button	Stop the proxy server.
Restart Button	Stops and restarts the proxy server. This is useful when you change/add/delete an include/exclude filter ex

The **include and exclude patterns** are treated as regular expressions (using Jakarta ORO). They will be matched against the h or implied) path and query (if any) of each browser request. If the URL you are browsing is

"http://jakarta.apache.org/jmeter/index.html?username=xxxx" ,

then the regular expression will be tested against the string:

"jakarta.apache.org:80/jmeter/index.html?username=xxxx" .

Thus, if you want to include all .html files, your regular expression might look like:

".*\.html(\?.*)?" - or **".*\.html"** if you know that there is no query string or you only want html pages without query strings.

If there are any include patterns, then the URL **must match at least one** of the patterns , otherwise it will not be recorded. If t
patterns, then the URL **must not match any** of the patterns , otherwise it will not be recorded. Using a combination of includ
should be able to record what you are interested in and skip what you are not.

N.B. the string that is matched by the regular expression must be the same as the **whole** host+path string.

Thus **"\.html"** will **not** match **j.a.o/index.html**

Versions of JMeter from 2.3.2 are able to capture binary POST data. To configure which content-types are treated as binary, u
property proxy.binary.types. The default settings are as follows:

```
# These content-types will be handled by saving the request in a file:
proxy.binary.types=application/x-amf,application/x-java-serialized-object
# The files will be saved in this directory:
proxy.binary.directory=user.dir
# The files will be created with this file filesuffix:
proxy.binary.filesuffix=.binary
```

It is also possible to have the proxy add timers to the recorded script. To do this, create a timer directly within the HTTP Prox
The proxy will place a copy of this timer into each sample it records, or into the first sample of each group if you're using grou
then be scanned for occurences of variable ${T} in its properties, and any such occurences will be replaced by the time gap fr
sampler recorded (in milliseconds).

When you are ready to begin, hit "start".

You will need to edit the proxy settings of your browser to point at the appropriate server and port, where the server is the machine JMeter is running on, and the port # is from the Proxy Control Panel shown above.

Where Do Samples Get Recorded?

JMeter places the recorded samples in the Target Controller you choose. If you choose the default option "Use Recording Con
stored in the first Recording Controller found in the test object tree (so be sure to add a Recording Controller before you start

If the Proxy does not seem to record any samples, this could be because the browser is not actually using the proxy. To check
stopping the proxy. If the browser still downloads pages, then it was not sending requests via the proxy. Double-check the bro
are trying to record from a server running on the same host, then check that the browser is not set to "Bypass proxy server for
example is from IE7, but there will be similar options for other browsers). If JMeter does not record browser URLs such as ht
http://127.0.0.1/, try using the non-loopback hostname or IP address, e.g. http://myhost/ or http://192.168.0.2/.

Handling of HTTP Request Defaults

If the HTTP Proxy Server finds enabled HTTP Request Defaults directly within the controller where samples are being stored
of its parent controllers, the recorded samples will have empty fields for the default values you specified. You may further con
placing an HTTP Request Defaults element directly within the HTTP Proxy Server, whose non-blank values will override tho
Request Defaults. See Best Practices with the Proxy Server for more info.

User Defined Variable replacement

Similarly, if the HTTP Proxy Server finds User Defined Variables (UDV) directly within the controller where samples are bei
within any of its parent controllers, the recorded samples will have any occurences of the values of those variables replaced by
variable. Again, you can place User Defined Variables directly within the HTTP Proxy Server to override the values to be rep
Practices with the Proxy Server for more info.

Please note that matching is case-sensitive.

Replacement by Variables: by default, the Proxy server looks for all occurences of UDV values. If you define the variable "W
"www", for example, the string "www" will be replaced by ${WEB} wherever it is found. To avoid this happening everywhe
Matching" check-box. This tells the proxy server to treat values as Regexes (using ORO).

If you want to match a whole string only, enclose it in ^$, e.g. "^thus$".

If you want to match /images at the start of a string only, use the value "^/images". Jakarta ORO also supports zero-width look
match /images/... but retain the trailing / in the output by using "^/images(?=/)". Note that the current version of Jakara ORO d
behind - i.e. "(?<=...) or (?<!...)".

If there are any problems interpreting any variables as patterns, these are reported in jmeter.log, so be sure to check this if UD
expected.

When you are done recording your test samples, stop the proxy server (hit the "stop" button). Remember to reset your browse
Now, you may want to sort and re-order the test script, add timers, listeners, a cookie manager, etc.

How can I record the server's responses too?

Just place a View Results Tree listener as a child of the Proxy Server and the responses will be displayed. You can also add a
file Post-Processor which will save the responses to files.

Cookie Manager

If the server you are testing against uses cookies, remember to add an HTTP Cookie Manager to the test plan when you have f
During recording, the browser handles any cookies, but JMeter needs a Cookie Manager to do the cookie handling during a te
Proxy server passes on all cookies sent by the browser during recording, but does not save them to the test plan because they a
between runs.

Authorization Manager

The Proxy server passes on any Authorization headers sent by the browser, but does not save them in the test plan. If the site r
you will need to add an Authorization Manager and fill it in with the necessary entries.

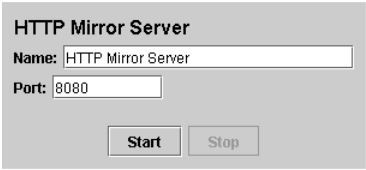
Uploading files

Some browsers (e.g. Firefox and Opera) don't include the full name of a file when uploading files. This can cause the JMeter p
One solution is to ensure that any files to be uploaded are in the JMeter working directory, either by copying the files there or
the directory containing the files.

18.9.6 HTTP Mirror Server

The HTTP Mirror Server is a very simple HTTP server - it simply mirrors the data sent to it. This is useful for checking the c
requests.

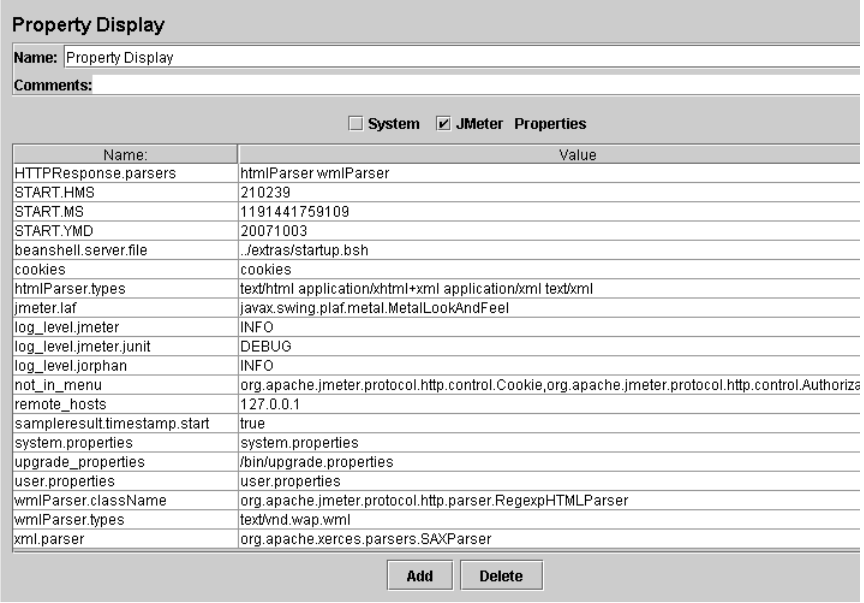
Control Panel



18.9.7 Property Display

The Property Display shows the values of System or JMeter properties. Values can be changed by entering new text in the Va
available only on the WorkBench.

Control Panel



Parameters

Attribute	Description	Required
Name	Descriptive name for this element that is shown in the tree.	No

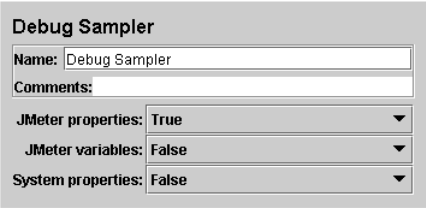
---

18.9.8 Debug Sampler

The Debug Sampler generates a sample containing the values of all JMeter variables and/or properties.

The values can be seen in the View Results Tree Listener Response Data pane.

Control Panel



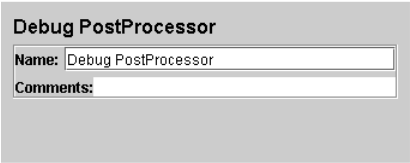
Parameters

Attribute	Description	Required
Name	Descriptive name for this element that is shown in the tree.	No
JMeter Properties	Include JMeter properties ?	Yes
JMeter Variables	Include JMeter variables ?	Yes
System Properties	Include System properties ?	Yes

18.9.8 Debug PostProcessor

The Debug PostProcessor creates a subSample with the details of the previous sampler properties. This is intended for develo

Control Panel



Parameters

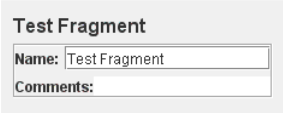
Attribute	Description	Required
Name	Descriptive name for this element that is shown in the tree.	No

---

18.9.9 Test Fragment

The Test Fragment is used in conjunction with the Include Controller and Module Controller .

Control Panel



Parameters

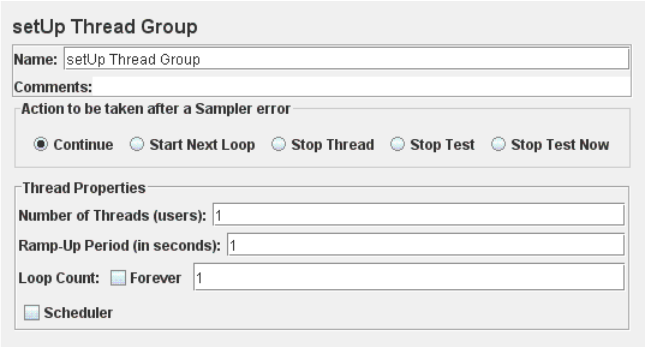
Attribute	Description	Required
Name	Descriptive name for this element that is shown in the tree.	Yes

---

18.9.10 setUp Thread Group

A special type of ThreadGroup that can be utilized to perform Pre-Test Actions. The behavior of these threads is exactly like a element. The difference is that these type of threads execute before the test proceeds to the executing of regular Thread Group

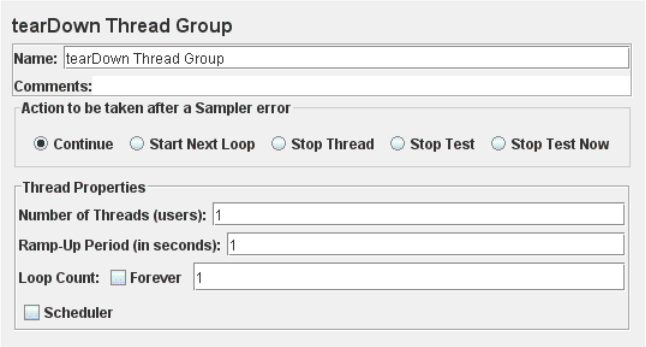
Control Panel



18.9.11 tearDown Thread Group

A special type of ThreadGroup that can be utilized to perform Post-Test Actions. The behavior of these threads is exactly like Group element. The difference is that these type of threads execute after the test has finished executing its regular Thread Gro

Control Panel



---

^

18.10 Reports

18.10.1 Report Plan

18.10.2 Report Table

18.10.3 HTML Report Writer

18.10.4 Report Page

18.10.5 Line Graph

18.10.6 Bar Chart

---

^

Index Next Prev
