

Introducción a JMeter: Conceptos Básicos

► Área: [Verificación de Entrega Software](#)

► Carácter del recurso: [Recomendado](#)

Código: RECU-0388

Tipo de recurso: Manual

Descripción

JMeter es una herramienta de testing cuyas funcionalidades se pueden resumir en tres:

- Diseñar un testplan, esto es, generar un fichero .jmx
- Ejecutar un testplan
- Ver de distintas formas los resultados de la ejecución de un testplan (vía listeners)

Para diseñar un testplan, JMeter dispone de una interfaz GUI a modo de diseñador, en la que el tester puede ir agregando componentes de manera visual, y ejecutar los componentes agregados, viendo el resultado. Una vez finalizado el diseño del testplan, la herramienta permite grabar este como un fichero .jmx.

La propia herramienta permite ejecutar un fichero .jmx previamente generado, vía línea de comandos o vía la propia interfaz GUI. La ejecución de un fichero .jmx realiza peticiones contra la aplicación objetivo a testear (peticiones del tipo que se hayan especificado al generar el fichero .jmx, JMeter dispone de la posibilidad de generar muchos tipos de peticiones: [HTTP](#), FTP, [LDAP](#), ...). Para cada petición ejecutada, JMeter recopila ciertos datos. Además, en el fichero .jmx se puede especificar que número de usuarios de cada tipo ejecuta las peticiones contra la aplicación, es decir, el .jmx simula una o mas comunidades de usuarios (roles) trabajando contra la aplicación objetivo.

Los datos generados por la herramienta para cada petición se procesan o bien con un tipo de componente que proporciona la interfaz GUI llamados listeners, o bien con herramientas externas. Los listeners permiten ver los resultados de una o mas ejecuciones de múltiples maneras (cada listener de una manera).

Este manual es una introducción al uso de la herramienta JMeter. Explica los conceptos básicos que es necesario conocer para *entender* la herramienta, y las posibles formas de utilizar ésta.

Es adecuado tanto para personas con un perfil técnico sin conocimientos previos que deseen iniciarse en el uso de la herramienta, como para directores, jefes de proyecto y analistas que deseen conocer las posibilidades de la herramienta en la ejecución de tests.

Los lectores interesados en profundizar en el uso de la herramienta quizás estén interesados en las siguientes referencias:

- [Construcción de un testplan con JMeter](#)
- [Ejecución de un testplan con JMeter](#)

Ventajas e inconvenientes

JMeter es una herramienta **ideal para realizar pruebas de rendimiento de aplicaciones web**. Sus principales debilidades son:

- **Aunque la versión actual (2.4) permite también el diseño y automatización de pruebas funcionales, existen otras herramientas más adecuadas para esta función.**
- JMeter NO se comporta como un navegador. Esto tiene varias implicaciones importantes: por defecto no guarda ni envía cookies, no interpreta código JavaScript, ... Cualquier funcionalidad de estas debe ser implementada específicamente en el testplan
- Con JMeter el tester trabaja a nivel de protocolos: el desarrollador de un testplan ha de descender a este nivel, por lo que normalmente el tester tiene que apoyarse en herramientas adicionales durante el desarrollo de un testplan, como [Firebug](#), [HttpFox](#), [SoapUI](#), [Badboy](#), ...
- Los tipos de aplicaciones que se pueden testear con JMeter dependen de los protocolos que implementen las interfaces de acceso a la aplicación. Con JMeter se pueden testear los siguientes tipos de interfaces: [HTTP](#), HTTPS, SOAP (sobre [HTTP](#)), XML-RPC (sobre [HTTP](#)), FTP, [LDAP](#), POP3, IMAP, SMTP, JMS, JDBC y TCP. El testeo de otro tipo de interfaces no es inmediato. En concreto, con JMeter no es posible (con un esfuerzo razonable), implementar un testplan para una interfaz RMI o WebDav

Prerequisitos e incompatibilidades de la tecnología

Los requisitos se especifican en la sección Instalación configuración, y acceso a la documentación.

No se han descrito incompatibilidades.

Si se va a hacer un uso masivo de esta herramienta, es importante conocer las posibilidades de la línea de comando y las propiedades de configuración para sacarle partido.

Relación con otros componentes y subsistemas

JMeter se utiliza principalmente en el subsistema de Verificación de [MADEJA](#), para medir el rendimiento de una aplicación. Sin embargo, no es descartable su uso en otros subsistemas, principalmente el de Desarrollo y Entorno para la automatización de tareas.

En el subsistema de Verificación, JMeter tiene aplicación en las siguientes áreas y pautas:

- Area: **Verificación de Entrega Software** > Pauta: **Realizar Pruebas Técnicas** > Subpauta: **Diseño de Pruebas Dinámicas**. En el subsistema de VERIFICACIÓN, esta pauta establece que se deben especificar los escenarios de pruebas, y grabar los casos de prueba que se vayan a realizar posteriormente en las pruebas de rendimiento. El apartado "Grabación de un testplan" explica como se pueden grabar casos de prueba con esta herramienta.
- Area: **Verificación y Ajustes en Entornos** > Pauta: **Realizar pruebas dinámicas**. En este área, esta pauta establece que entre otros aspectos, "... se pretende verificar si el rendimiento de la aplicación es bueno ante situaciones de carga y estrés...", y establece cuatro tipos de pruebas de rendimiento: rendimiento, estrés, sostenidas y picos. La mayoría de los apartados de este documento son aplicables a esta pauta, pero especialmente "Ejecución de un testplan. Captura y procesamiento de la información" y "Medición del rendimiento. Conceptos estadísticos".

Existen algunas herramientas complementarias de JMeter para realizar determinadas tareas:

- **Badboy©**

Badboy© es una herramienta de [Badboy Software©](#)

Es una herramienta de testing de aplicaciones web, que funciona con Windows© e Internet Explorer©. Su filosofía consiste en ser un cliente de automatización de Internet Explorer© (utiliza el API de éste como motor de navegación).

Básicamente lo que hace la herramienta es grabar como un script (en un fichero con extensión .bb) la secuencia de navegación por la web, que el usuario realiza con la propia herramienta. Una vez generado el fichero .bb, la misma secuencia de navegación se puede volver a reproducir ejecutando el script .bb con la propia herramienta. La herramienta permite además modificar el script grabado introduciendo comprobaciones, condiciones, ...

Badboy© utiliza su propio formato de fichero (los .bb internamente no tienen nada que ver con los .jmx), por lo que un script grabado con Badboy© no es ejecutable con JMeter. Sin embargo, Badboy© dispone de la funcionalidad de grabar en formato .jmx una secuencia de navegación capturada. Esto es lo que lo hizo popular entre los usuarios de JMeter.

Para grabar como .jmx un script de Badboy, basta seleccionar la opción File / Export to JMeter ... en la interfaz GUI de Badboy©.

En la web de Badboy© (<http://www.badboy.com.au/>) puede obtener una distribución de la herramienta y acceder a su documentación online.

- **Firebug**

Es un plug-in de Firefox: <http://getfirebug.com/>. Permite inspeccionar fácilmente el código HTML y XML devuelto por la aplicación objetivo, en el proceso de implementación del testplan (fichero .jmx)

- **SoapUI**

Es una herramienta GUI para probar servicios web de tipo SOAP: <http://www.soapui.org/>

Se utiliza para copiar las cadenas XML de peticiones SOAP y pegarlas en el testplan de JMeter, cuando este testea la interfaz SOAP de una aplicación objetivo.

- Automatizar ejecuciones de JMeter vía ant / maven / Hudson.

Toda la interacción con JMeter se puede automatizar vía ant o maven, y con estos vía Hudson. Algunas referencias:

- Plugin de maven para JMeter: <http://wiki.apache.org/jakarta-jmeter/JMeterMavenPlugin>
- Tarea de ant para Jmeter, que se incluye también en el directorio \${JMeter}/extras (ant-jmeter-1.0.9.jar) y build.xml de ejemplo: <http://www.programmerplanet.org/pages/projects/jmeter-ant-task.php>
- Plugin de Hudson para JMeter: <http://wiki.hudson-ci.org/display/HUDSON/Performance+Plugin>

- Generar un .html explicativo de un testplan (fichero .jmx).

Directorio \${JMeter}/extras: el script de ant schematic.xml permite aplicar la hoja de estilos schematic.xsl a un .jmx, para generar a partir de él un .html

- Generar un .html explicativo de un fichero result sample.

Directorio \${JMeter}/extras: el script de ant build.xml tiene una tarea report que aplica la hoja de estilos jmeter-results-detail-report_21.xsl al .jtl que se genera en la ejecución. La propiedad jmeter.save.saveservice.xml_pi (ver jmeter.properties) permite añadir una instrucción de procesamiento al .jtl que genera JMeter, por ejemplo para aplicar una hoja de estilos al XML.

Modo de empleo

Instalación, configuración, y acceso a la documentación

Descargar el software de <http://jakarta.apache.org/jmeter/> (sitio oficial del proyecto).

Licencia: [Apache License Version 2.0](#)

Prerrequisitos: JRE 1.5+

Instalación:

- Basta con descomprimir en un directorio (por ejemplo en /usr/local/ en Unix/Linux, o C: en Windows), el paquete de distribución del software (.zip o .tgz).
- Aunque para usar JMeter no es imprescindible, se recomienda crear una variable de entorno (por ejemplo de nombre JMETER_HOME), cuyo valor sea al path absoluto al directorio de instalación. En lo que sigue, asumiremos que esta variable ha sido creada. Si por alguna razón, esta variable no existe, sustituir las expresiones %JMETER_HOME% (Windows) ó \${JMETER_HOME} por el path absoluto al directorio raíz de la instalación de JMeter, aquellos sitios donde las utilicemos.
- Aunque para usar JMeter tampoco es imprescindible, se recomienda descargar e instalar los plugins para JMeter del proyecto [JMeter Plugins](#). Estos plugins añaden componentes adicionales a los que JMeter implementa por defecto. Descarga e instrucciones de instalación en el sitio del proyecto.

Comprobación de la instalación:

```
${JMETER_HOME}/bin/jmeter.sh --version
```

debería mostrar un texto como el siguiente:

```
Copyright (c) 1998-2010 The Apache Software Foundation
Version 2.4 r961953
```

Configuración:

La ejecución de JMeter se configura vía ficheros de propiedades o vía opciones de línea de comando (información sobre esto más adelante). El fichero jmeter.properties establece la mayoría de las propiedades de configuración.

Hay un aspecto de la configuración que merece cierta atención: en la interfaz GUI se puede consultar ayuda contextual sobre un componente de un testplan haciendo click en la opción Ayuda> del menú emergente. JMeter busca la ayuda contextual por el nombre en Inglés del tipo de componente. Si la interfaz GUI está configurada en Español, los nombres de los tipos de componentes figuran traducidos, por lo que no le es posible encontrar la ayuda. Por esta razón, y porque el manual de usuario del producto está en Inglés, se recomienda configurar la interfaz GUI en idioma Inglés. Para ello, establecer en el fichero \${JMETER_HOME}/bin/jmeter.properties la propiedad language:

```
language=en
```

Ejecutar JMeter (la interfaz GUI):

```
${JMETER_HOME}/bin/jmeter.sh
```

Más información sobre la interfaz GUI en los puntos que siguen.

Documentación:

La documentación online de la última versión del producto puede consultarse en <http://jakarta.apache.org/jmeter/usermanual>. Los apartados del manual [18. Component reference](#) y [19. Functions](#) se utilizan muy frecuentemente como referencia durante la construcción de un testplan.

Utilización de la herramienta

Una vez instalada la herramienta, la forma más habitual de utilizarla es:

1. **Grabar las interacciones de cada uno de los tipos de usuarios con la aplicación software.** Esto genera un caso de pruebas para cada una de las tipologías de usuarios (perfiles) que admite la aplicación. Posteriormente, cada uno de los casos de prueba se ejecutarán de manera concurrente por parte de N usuarios por cada caso de prueba.
2. **Adaptar el script (fichero .jmx) generado en el paso anterior.** Por ejemplo, para que sea configurable, para que se hagan comprobaciones que no se pueden generar durante la grabación, ...
3. **Comprobar que el script realiza correctamente las tareas que se desea que realice el caso de prueba.**
4. **Integrar todos los scripts**
5. **Ejecutar el testplan**
6. **Visualizar los resultados de la ejecución**

En los apartados que siguen explicamos los conceptos de funcionamiento de JMeter relativos a estos cinco puntos. En este manual solo explicaremos los conceptos y posibilidades de la herramienta. Ver [Construcción de un testplan](#) con JMeter para más detalle sobre los cuatro primeros puntos, [Ejecución de un testplan con JMeter](#) para más detalles sobre los dos últimos.

Como funciona JMeter

La interfaz GUI de JMeter nos permite construir un testplan. Más sobre la interfaz GUI en el apartado Interfaz GUI, más sobre la construcción de un testplan en los apartados [Grabación de un testplan](#) y Construir planes configurables y mantenibles....

Una vez construido un testplan, este consiste en un fichero con extensión (por defecto) .jmx . Más sobre los ficheros .jmx en el

apartado Testplan (*.jmx).

Un testplan se puede ejecutar desde la línea de comando o desde la interfaz GUI. Más sobre la sintaxis de línea de comando en el apartado [Opciones de línea de comando](#). Más sobre la ejecución de un testplan en el apartado [Modos de ejecución de JMeter](#) y [Ejecución de un testplan](#). Captura y procesamiento de la información.

Brevemente, lo que hace un testplan durante su ejecución es crear el número de *threads* que se especifican en él. Cada thread representa a un usuario (virtual), que realiza con la *aplicación objetivo* a testear, las interacciones que se especifican en el testplan. Todos los usuarios (threads) de un mismo grupo (los threads se pueden agrupar, véase el componente *Thread Group*), realizan las mismas interacciones con la aplicación. Las interacciones de los usuarios virtuales con la aplicación las realizan un *tipo de componente* del testplan denominados *samplers* que son los que hacen peticiones a la *aplicación objetivo* a testear. Existen samplers de diferentes tipos, para hacer peticiones [HTTP](#), peticiones FTP, peticiones [LDAP](#), ... Más sobre componentes y tipos de componentes de un testplan en el apartado [Tipos de componentes de un testplan, ámbito y orden de ejecución](#).

Durante la ejecución de un testplan, JMeter captura varios datos sobre **todas y cada una** de las peticiones realizadas por los samplers del testplan a la aplicación objetivo. Entre los datos capturados para cada petición están: timestamp de la petición, identificación del grupo de threads y thread que realiza la petición, URL de la petición, resultado de la petición, tiempo de respuesta, ... La forma en que se capturan las peticiones realizadas por los samplers es en ficheros, bien en formato CSV o bien en formato XML (ficheros .jtl). Estos ficheros se llaman *result samples* o *sample result* o simplemente *ficheros sample* o *ficheros result*. Más sobre los ficheros result samples en Ficheros sample result y sobre la captura de información por JMeter en el apartado [Ejecución de un testplan. Captura y procesamiento de la información](#).

De la lista de peticiones capturadas durante la ejecución de un testplan pueden extraerse muchas *series estadísticas*. Estas series estadísticas pueden procesarse para calcular diferentes magnitudes. Así por ejemplo, si la serie estadística consiste en el tiempo de respuesta de cada petición, pueden hallarse varias magnitudes sobre el tiempo de respuesta de la aplicación: la media, el máximo, el mínimo, la mediana, la desviación estándar, el percentil 90, ... Más sobre esto en [Medición del rendimiento. Conceptos estadísticos](#).

Por otra parte, en JMeter existe un tipo de componente denominado *listener*. Los listeners son componentes que permiten procesar la información capturada por JMeter durante la ejecución de un testplan. Cada tipo de listener procesa la información capturada de una manera: unos la muestran en forma de tabla calculando diferentes magnitudes estadísticas, otros la muestran en forma de gráfica, otros simplemente salvan la información previamente filtrada a un fichero, ... La clave es que todos los listeners independientemente de como procesen la información, parten de la misma información base: los datos de todas y cada una de las peticiones capturadas durante la ejecución de un testplan. Los listeners se suelen utilizar de dos maneras:

- Una vez capturadas las peticiones del testplan en un fichero .csv ó .jtl, el fichero se puede cargar en el listener (en la interfaz GUI), y ver la información capturada de la forma como la muestra el listener.
- En la interfaz GUI, durante la ejecución de un testplan los listeners muestran en tiempo real el resultado de procesar la información que se va capturando.

Más sobre listeners en el apartado [Ejecución de un testplan. Captura y procesamiento de la información](#).

Artefactos software

En este apartado enumeraremos y explicaremos en que consisten los diferentes tipos de artefactos software relacionados con JMeter:

Ficheros de propiedades (*.properties)

JMeter se basa en varios ficheros de propiedades que configuran diferentes aspectos de la ejecución.

El principal es \${JMeter_HOME}/bin/jmeter.properties, en el que se especifica la mayoría de las propiedades de configuración. Se aconseja consultar los comentarios de este fichero, para información sobre las posibilidades de configuración.

Hay varias propiedades de este fichero que referencian a otros ficheros de propiedades (de nombre configurable), que también se leen durante el arranque de JMeter. Las mencionamos con sus valores por defecto:

- httpclient.parameters.file=httpclient.parameters
- user.properties=user.properties
- system.properties=system.properties

Para cambiar la configuración de JMeter vía ficheros de propiedades, editar el fichero .properties y rearrancar JMeter.

Ejecutables de línea de comando

En el directorio \${JMeter_HOME}/bin/:

- jmeter, jmeter.sh, jmeter.bat
Ejecutan JMeter desde línea de comando o la interfaz GUI. Más información sobre la línea de comando en el apartado sobre opciones de línea de comando, y sobre la interfaz GUI en el punto que sigue.
- jmeter-server, jmeter-server.bat
Ejecutan JMeter en modo servidor (ver modo de ejecución)
- shutdown.sh, shutdown.cmd
Para la ejecución de un testplan de forma no abrupta
- stoptest.sh, stoptest.cmd

Idem, de forma abrupta

- mirror-server, mirror-server.cmd

JMeter dispone de un servidor [HTTP](#) embebido que se puede usar con propósitos de depuración durante la construcción de un testplan (ver componente [HTTP Mirror Server](#) en la [documentación de componentes](#))

- jmeter-*.cmd

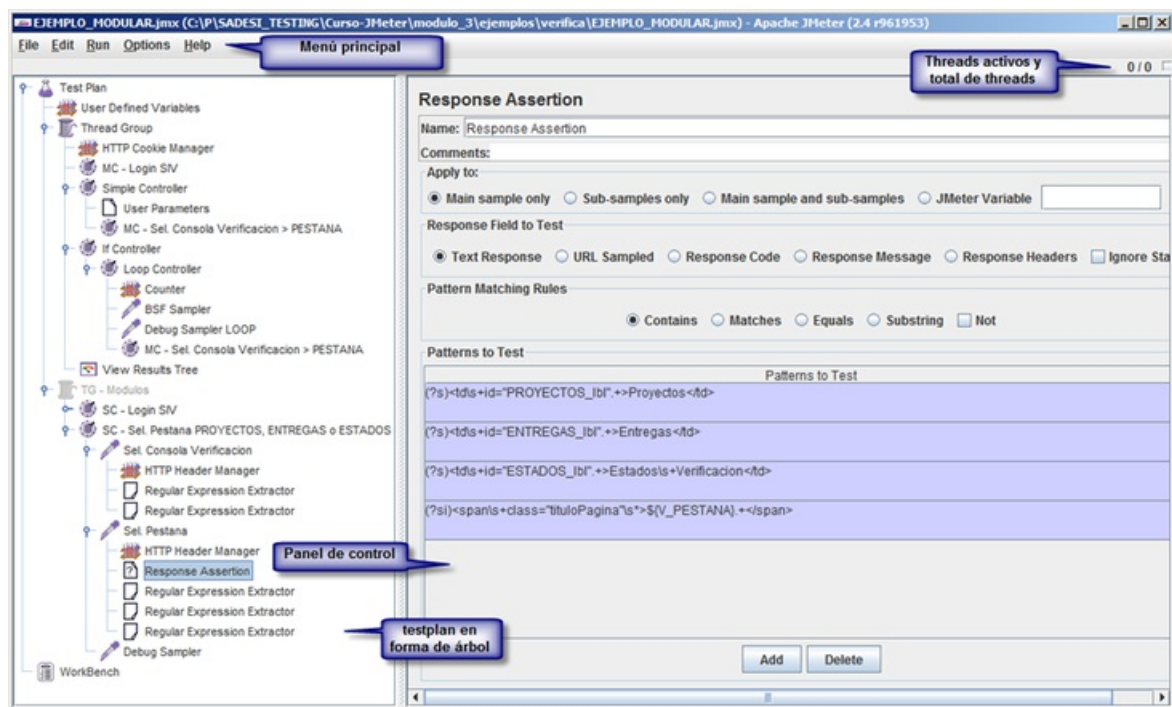
Ejecutables específicos de Windows para ejecutar un testplan arrastrando y soltando.

Interfaz GUI

JMeter se puede ejecutar de dos formas: vía línea de comando o vía la interfaz GUI. De las dos maneras podemos ejecutar un testplan. Pero además, la interfaz GUI sirve para diseñar un testplan.

Por defecto los ejecutables del directorio `${JMMETER_HOME}/bin/` arrancan JMeter en modo GUI. Si se desea sólo ejecutar un testplan desde línea de comando sin mostrar la interfaz GUI, hay que usar los modificadores `-n` y `-t` (información sobre estos modificadores en el apartado Opciones de línea de comando).

La imagen siguiente muestra el aspecto de la interfaz GUI para la versión actual de JMeter (2.4):



En el frame de la izquierda se muestra un árbol que constituye el testplan. Este árbol es la forma en que JMeter nos muestra el testplan.

Cada nodo del árbol es un *componente* que el tester añade al árbol en tiempo de construcción del testplan.

Cada componente es de un *tipo de componente* concreto, y según el tipo del componente, al seleccionarlo se muestra un panel de control diferente en el frame de la derecha. En la imagen se ha seleccionado un componente de nombre "Response Assertion", por lo que en el frame de la derecha se muestra el panel de control de este tipo de componente.

Más información sobre los tipos de componentes y el orden en que se ejecutan los componentes de un testplan, en el apartado [Tipos de componentes de un testplan](#), ámbito y orden de ejecución.

Editando los campos del panel de control de cada componente es como se configura el comportamiento de éste en el testplan.

Hay tres campos que existen en cualquier panel de control independientemente del tipo de componente: el tipo del componente ("Response Assertion" en la imagen), campo "Name" (que por defecto es el mismo que el tipo del componente), y campo "Comments" (para explicar lo que se necesite sobre el componente). El tester puede poner cualquier texto en los dos últimos al construir el testplan.

La forma en que se añade un nuevo componente a un nodo del árbol una vez seleccionado es mediante la opción `Edit > Add` ó `Edit > Insert Parent`, o bien mediante el menú emergente del componente seleccionado.

Los nodos del árbol admiten operaciones de Copiar, Cortar, Pegar, Eliminar, Activar/Desactivar, Mover (arrastrando y soltando), ...

Físicamente en disco, un testplan es un fichero `.jmx`: cuando el tester graba el testplan JMeter genera un fichero `.jmx` (en la barra de título se muestra el fichero que tenemos abierto en este momento).

El menú `Run` tiene opciones para ejecutar el testplan y parar la ejecución. Durante la ejecución de un testplan se muestran dos números (N/M) en la esquina superior derecha. N es el número de threads activos en cada momento, y M es el número total de threads que crea el testplan (más sobre threads en el apartado Como funciona JMeter).

Testplan (*.jmx)

Físicamente un testplan es un fichero `.jmx` (extensión por defecto), generado con la interfaz GUI de JMeter. Internamente, los

ficheros .jmx son ficheros de texto en formato XML (por lo que podemos editarlos con un editor de textos cualquiera, la interfaz GUI nos proporciona un "editor" especializado en este tipo de ficheros).

Un testplan (fichero .jmx) implementa uno o varios casos de prueba. JMeter no impone ninguna restricción en cuanto a la atomicidad de los casos de prueba respecto a los ficheros .jmx, por lo que ésta es una decisión del implementador (esto es, del tester).

Se puede ejecutar un testplan de dos formas: desde la línea de comando o desde la interfaz GUI (véase [Opciones de línea de comando](#) y [Modos de ejecución de JMeter](#)).

Fichero de log

Como casi todas las aplicaciones Java, JMeter genera un fichero de log durante su ejecución. Por defecto, el fichero de log se genera en el directorio actual desde el que se ejecuta JMeter, y con el nombre por defecto jmeter.log .

Si no se desea que cada ejecución de JMeter sobrescriba el fichero de log generado en la ejecución anterior, se puede especificar que el nombre del fichero contenga la fecha y hora de la ejecución. Esto se puede hacer:

- O bien editando la propiedad log_file de jmeter.properties, por ejemplo:

```
log_file='jmeter_'yyyyMMddHHmmss'.log'
```

- O bien especificando el formato de nombre de fichero de log en la línea de comando con el modificador -j :

```
${JMeter_HOME}/bin/jmeter -j 'jmeter_'yyyyMMddHHmmss'.log'
```

Ficheros sample result

Durante la ejecución de un testplan, JMeter captura varios datos sobre **todas y cada una** de las peticiones realizadas por los samplers del testplan a la aplicación objetivo. Estas peticiones (y opcionalmente la respuesta a las mismas por parte de la aplicación), se pueden salvar en ficheros mientras se está ejecutando un testplan. Estos ficheros se llaman *sample result*, *result sample*, ficheros *sample* o simplemente ficheros *result* (de resultados).

JMeter proporciona dos procedimientos para generar este fichero en cada ejecución:

- Especificando en la línea de comando la opción -l. Por ejemplo:

```
${JMeter_HOME}/bin/jmeter -l 'jmeter_'yyyyMMddHHmmss'.csv'
```

(la opción -l también permite especificar el formato de hora como parte del nombre del fichero).

- Especificando en un listener del testplan, un path absoluto a un fichero en el campo Filename: del listener (vía la interfaz GUI). Todos los listeners tienen un campo Filename: en su panel de control.

El formato de los ficheros de sample result es o bien CSV o bien XML. El formato del fichero se establece de una de esta dos formas:

- Modificando la propiedad jmeter.save.saveservice.output_format de jmeter.properties. Con esta forma se establece el formato por defecto para todos los ficheros de este tipo. Esto se puede hacer en la línea de comando sin necesidad de cambiar el fichero jmeter.properties, con el modificador -J. Por ejemplo:

```
${JMeter_HOME}/bin/jmeter -Jjmeter.save.saveservice.output_format=csv -l 'jmeter_'yyyyMMddHHmmss'.csv'
```

- En el cuadro de diálogo de configuración del listener, que se muestra haciendo click en el botón Configure... (todos los listeners tiene también un botón Configure... que al ser pulsado muestra un cuadro de diálogo en el que se pueden establecer diversas opciones de configuración del fichero especificado en Filename:).

Notas:

- JMeter no comprueba la correspondencia entre el formato que se especifique para el fichero (CSV o XML), y la extensión que se utilice para el nombre de éste. Por tanto, es posible generar un fichero XML con extensión .csv (cuidado con esto). Normalmente se utiliza extensión .jtl cuando el formato del fichero es XML.
- Las respuestas a las peticiones solo se pueden grabar en el fichero result sample si el formato de éste es XML.
- En pruebas de rendimiento se desaconseja el uso del formato XML, en favor del CSV, debido a JMeter consume menos recursos para generar este último. Así mismo, se desaconseja el capturar las respuestas de la aplicación en este tipo de pruebas.
- Una nuevas funcionalidades previstas para JMeter es poder guardar la información de estos ficheros (peticiones) directamente en una tabla de base de datos.

Los campos de información que se capturan para cada petición son prácticamente los mismos para el formato CSV y XML. En el manual de usuario de JMeter (secciones [14.4 CSV Log format](#), [14.5 XML Log format 2.1](#) y [14.7 Sample Attributes](#)) se explica el significado de cada uno de estos campos en ambos formatos.

La equivalencia entre campos de ambos formatos es:

Campo CSV	Atributo XML
=====	=====
timeStamp	ts

elapsed	t
label	lb
responseCode	rc
responseMessage	rm
threadName	tn
dataType	dt
success	s
failureMessage	--
bytes	by
grpThreads	ng
allThreads	na
URL	--
Filename	--
latency	lt
encoding	de
SampleCount	sc
ErrorCount	ec
Hostname	hn
IdleTime	it
Variables	vamame

Listeners

Los listeners son uno de los tipos de componentes de un testplan (más sobre los tipos de componentes de un testplan en Interfaz GUI y Tipos de componentes de un testplan, ámbito y orden de ejecución).

La importancia que tienen los listeners respecto a los otros tipos de componentes es que procesan las peticiones realizadas a la aplicación objetivo por los samplers del testplan, y muestran el resultado del procesamiento de diferentes formas según el tipo de listener de que se trate.

Se pueden usar los listeners de dos formas:

- Si el testplan se ejecuta desde la interfaz GUI, muestran en tiempo real el procesamiento de las peticiones que realizan los samplers. Así por ejemplo, el listener View Results Tree muestra los samplers en el orden en que se ejecutan, y para cada uno de ellos se puede visualizar los datos de la petición realizada y la respuesta de la aplicación (muy útil para depurar un testplan mientras se construye). Otro ejemplo: el listener Graph Results muestra en forma de gráfica los tiempos de respuesta de la aplicación objetivo.
- Una vez ejecutado el testplan, si se ha generado el fichero sample result, este se puede cargar en el campo Filename: de un listener (por ejemplo creando un plan vacío que solo contenga el listener que nos interesa, o creando el componente listener que nos interesa en el componente Workbench de la interfaz GUI). Al hacer esto, el listener mostrará el resultado de procesar el fichero en un determinado formato.

Nota: Esta última es la forma como se recomienda trabajar en la ejecución de pruebas de rendimiento. Resumiendo,

- Generar el fichero sample result en formato CSV, vía opción -l de la línea de comando o vía un listener global del tipo Simple Data Writer.
- Una vez ejecutada la prueba de rendimiento, cargar el fichero result sample generado en uno o varios de los listeners que nos muestren las medidas de rendimiento que queremos obtener.

Características

JMeter implementa las siguientes funcionalidades:

- Simular, con el nivel de concurrencia que se desee, las interacciones de una o varias comunidades de usuarios virtuales (perfiles o roles) con la aplicación
- Interfaz GUI para diseñar casos de prueba (testplans)
- Posibilidad de grabar las interacciones de un usuario, vía navegador, con la interfaz [HTTP](#) de una aplicación, y convertir la grabación en un testplan
- Ejecución de un testplan desde la interfaz GUI o desde línea de comando, permitiendo así la automatización de la ejecución de pruebas.
- Construcción de testplans configurables
- Ejecución de un testplan en modo cliente, en modo remoto o en modo distribuido
- Visualización de los resultados de la ejecución de un testplan de múltiples formas, obteniendo diferentes métricas.

Versiones

Versión actual en el momento de redactar este documento, y por tanto utilizada como referencia para los conceptos que se

explican: 2.4

Para más información sobre versiones e histórico de versiones ver la [página de descarga de JMeter](#).

Para migrar un script .jmx de la versión 2.3 a la 2.4, simplemente abrirlo con la interfaz GUI y volver a salvarlo.

Buenas prácticas y recomendaciones de uso

- Permitir que los scripts de JMeter (ficheros .jmx) sean configurables
Datos como direcciones IP, nombres de host, puertos, número de usuarios, tiempo de ejecución del testplan o número de iteraciones deberían ser propiedades configurable que el usuario pueda establecer antes de la ejecución del testplan. De esta forma los scripts .jmx no son específicos de los entornos en los que se despliegan las aplicaciones que testean
- No usar listeners innecesarios en el testplan
El tiempo extra que los listeners emplean en procesar cada petición a la que afecta el listener, se contabiliza como parte del tiempo de ejecución de la petición. Esto influye negativamente en scripts .jmx que testean el rendimiento
- Capturar el fichero de sample result, y el fichero de log de cada ejecución en ficheros específicos, que contengan la fecha-hora como parte del nombre
- En pruebas de rendimiento, capturar el fichero de sample result en formato CSV
El formato JTL contiene más texto porque es XML, por lo que JMeter requiere más tiempo para generarlos. Esto impacta negativamente en la ejecución de tests de rendimiento
- Documentar la secuencia de interacciones que realiza un script .jmx generando un documento externo, por ejemplo un fichero de texto. De otra forma es muy difícil que el script sea modificable por quien lo haya generado. Esto cobra más importancia cuanto más complejas sean las interacciones que el script realiza con la aplicación

Ejemplos

El fichero adjunto JMeterejemplo.zip contiene un testplan configurable para la aplicación eCO, así como varios ficheros de log y de sample result correspondientes a otras tantas ejecuciones. Ver el fichero LEAME.txt para más información

Enlaces externos

- [Web oficial de la herramienta](#)
- [Artículos](#)
- [Manual de usuario de JMeter](#)
- [SoapUI](#)
- [Badboy](#)
- [Firebug](#)
- [Plugin de Maven para JMeter](#)
- [Plugin de Hudson para JMeter](#)
- [Varios plugins para JMeter](#)

Source URL: <http://madeja.i-administracion.junta-andalucia.es/servicios/madeja/contenido/recurso/388>