

Configuration

by Ovidiu Predescu, Jeff Turner

NOTICE: Copyright © 2002-2003 Ovidiu Predescu and Jeff Turner. All rights reserved.
The Ant eater manual may be reproduced and distributed in whole or in part, in any medium, physical or electronic, so long as this copyright notice remains intact and unchanged on all copies.

1. Configuration

Anteater is configured through a set of properties. Properties belong to groups, and are named as such. Thus, `default.debug` is the *debug* property in the `default` group. If you have yet to read the [Grouping](#) section, now would be a good time to do so. The main thing to remember is that all tasks and groups belong to the `default` group, unless explicitly overridden, and thus properties in the `default` group will be inherited in subsequent groups unless overridden.

Currently defined default properties are:

Property name	Type	Default value	Description
<code>default.host</code>	String	<code>localhost</code>	Set the default host for httpRequest or soapRequest requests.
<code>default.port</code>	Integer	<code>BUILTIN,8080</code>	Set the default port to use. The preceding <code>BUILTIN</code> means use the <code>ServletContext</code> default, and if not available, fall back to the indicated port.
<code>default.debug</code>	Integer	<code>0</code>	Default debug level. 0 is

			lowest, 10 highest. The higher the debug level, the greater the number of information being logged.
default.session	boolean	true	Whether the default group defines a session element.
default.timeout	Integer	30s	How long the client-side action tasks should wait before assuming a server is dead. If the request times out it is consider to have failed.
protocol	String	HTTP/1.0	HTTP protocol to connect as. If virtual hosts are used, must be HTTP/1.1.
default.halt.onerror	boolean	true	Whether a failed action task halts the Ant eater test script.
default.usetidy	boolean	false	<p>Whether to apply JTidy on the result obtained by action tasks (like HttpRequest or soapRequest) to clean up mark-up text and transform it in valid XML.</p> <p>To be able to apply matchers such as xpath in your tests, the content to be matched must be valid XML. If you know the response is an HTML document which is not valid XML, you should set this flag to true to be able to use xpath on it.</p>

Configuration

			<p>One caveat with transforming HTML into XML is that the structure of your document might change. In order to correctly apply XPath on the resulting document, you need to inspect the result manually. Antecater tries to anticipate JTidy's changes by automatically setting the 'ignoreSpaces' and 'singleLine' attributes of xpath and regexp to true.</p>
default.usetidy-server	boolean	false	<p>Whether to apply JTidy on the content body received in a request by the listener task. This is very similar with the effect of <code>default.useTidy</code>, except that its action happens on the incoming requests, rather than on the result obtained by action tasks.</p>
default.filename-format	String	(see description)	<p>Flag specifying the default filename format for log files. Default value is <code>TEST-\${groupid}_\${taskname}_\$</code></p> <p>Pretty much any property can be used, both Antecater-specific properties (e.g. the task's description), Group properties, and Ant <code><property></code> properties.</p> <p>There is one quirk in the format: variables of the form <code>\${prefix:variable}</code>. These are interpreted as</p>

Configuration

			<p>follows: if <code>\${variable}</code> is defined, and has value <code>value</code>, then <code>\${prefix:variable}</code> is replaced with <code>'prefixvalue'</code>. For example, <code>\${run_:run}</code> becomes <code>'run_1'</code>, or <code>\${run at :date}</code> becomes <code>'run at 10/3/03'</code>. If <code>variable</code> is undefined, the variable is replaced with <code>"</code>. This hackery is primarily for the <code>'run'</code> variable, which won't exist if <code>overwrite</code> is true (see below).</p>
<code>default.overwrite</code>	boolean	true	<p>Flag indicating whether, by default, we should overwrite log files from previous Anteater runs.</p> <p>If false, log files are made unique with a <code>runX</code> filename suffix, where <code>X</code> is incremented to ensure a unique filename.</p>
<code>default.enable</code>	boolean	true	<p>This flag indicates whether member tasks will be run or not.</p> <p>For example, one might classify action tasks into <code>'normal'</code> and <code>'strict'</code> groups, and then at runtime, choose to disable one or the other group, eg <code>-Dstrict.enable=false</code> to turn off the <code>'strict'</code> group's tasks.</p>