

# Table of Contents

<b>1 Plugin structure.....</b>	<b>1</b>
1.1 Properties used by plugins.....	1
1.1.1 General properties.....	1
1.1.2 Language specific properties.....	1
1.1.3 Plugin specific properties.....	1
1.2 Methods called by plugin framework.....	1
1.2.1 Register.....	1
1.2.2 Execute.....	1
1.2.3 Publish.....	1
1.2.4 Finalise.....	2

# 1 Plugin structure

## 1.1 Properties used by plugins

### 1.1.1 General properties

notest	disables all plugins registered at TEST target
no{ plugin name }	disable a specific plugin
src.location	sets the location of source files (any language) analysed by a plugin

### 1.1.2 Language specific properties

java.src.location	sets the location of java source files
java.class.location	sets the location of java class files
java.jar.location	sets the location of java jar files
java.lib.location	sets the location of java lib files

### 1.1.3 Plugin specific properties

Plugin specific properties can be used to set a threshold or allow for any other customisation that may be necessary for the plugin. These properties should start with <plugin.name>. For example, 'findbugs.class.location' or 'findbugs.failure.threshold'.

## 1.2 Methods called by plugin framework

### 1.2.1 Register

```
def register(self, pluginManager):
```

Is called once to register the plugin. It sets where the plugin should be executed, normally during 'pretest'. Here the plugin also verifies that the plugin is correctly installed, i.e. it has the files it needs. If it does not find any missing files, it calls

```
self.pluginManager.register(self.__module__, definition)
```

### 1.2.2 Execute

```
{def execute(self, profile=None, command=None, target=None, targetString=None, propertyManager=None, bu
```

Is called for each component being built.

Runs the metric application, for example with an apache-ant command.

### 1.2.3 Publish

```
def publish(self, profile=None, command=None, target=None, targetString=None, propertyManager=None, bu
```

Is called for each component being built.

In this function the plugin publishes metric data for each component and links to the detailed html/xml pages created by the metrics application.

To creates the module metrics, the framework utilities should be used, as follows:

## EticsPluginsDevel < EMI < TWiki

```
metrics = buildStatus.CreateMetrics('Plugin name')
metrics.detailshtmllink = #path to component specific .html page (created by metrics application,
metrics.detailsxmllink = #path to component specific .xml page (created by metrics application, n
metrics.values = #array with all the values
metrics.value = #overall value
metrics.type = #integer, float, percent, etc.
metrics.unit = #bugs, violations, etc.
buildStatus.setModuleMetrics (propertyManager ['moduleName'], propertyManager ['name'], propertyManage
```

This will introduce the results as xml fields in the build status xml, which contains all build data and is sent to the repository, making it possible to later get them by querying the repository.

### 1.2.4 Finalise

```
def finalise (self, profile=None, command=None, target=None, targetString=None, propertyManager=None, bu
```

Is called once to finalise the plugin and publish the overall results.

First it should create the plugin specific report page from a template file. It replaces lines in the template .html file with html created by the plugin.

Then it creates the summary page located in the /reports/index.html page in the report where it shows the metric data as an average of each component's data. Once again it should also use the framework functions to introduce the overall results in the build status xml:

```
metrics = buildStatus.CreateMetrics('Plugin name')
metrics.detailshtmllink = 'plugin name/index.html' #path to plugin specific .html page.
metrics.values = #array with all the values
metrics.value = #overall value
metrics.type = #integer, float, percent, etc.
metrics.unit = #bugs, violations, etc.
buildStatus.setOverallMetrics (metrics)
```

-- FabioCapannini - 06-Sep-2011

---

This topic: EMI > EticsPluginsDevel  
Topic revision: r4 - 2011-11-16 - DuarteMeneses



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback