

# AutomaturePlugin

## Plugin Information

No information for the plugin 'automature-reporter' is available. It may have been removed from distribution.

Adds the ability to connect to Zermatt server and upload test execution results from a xunit formatted XML results file.

## Description

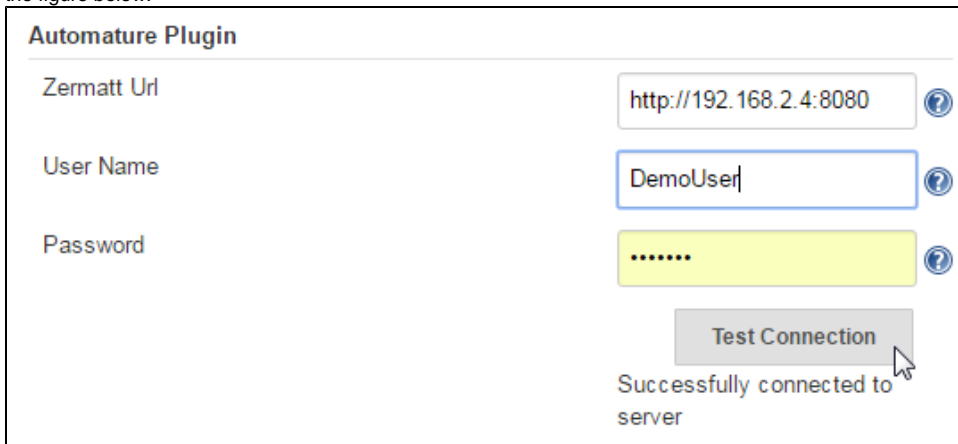
This Plugin provides the ability to connect to Zermatt server and upload test execution results from a xunit formatted XML results file.

## Installation

- Navigate to [Manage Jenkins](#) > [Manage Plugins](#)
- Click Advanced tab
- Upload automature-reporter.hpi file

## Set Up

- Navigate to [Manage Jenkins](#) > [Configure System](#) > [Automature Plugin](#)
- **Zermatt URL**
  - Provide address of Zermatt server. If Zermatt is running at <http://zermatt.automature.com> and port 8080 then the url should be zermatt.automature.com:8080
- **User Name**
  - Provide your Zermatt Username
  - The user must exist in Zermatt
- **Password**
  - Provide your Zermatt Password
- **Test Connection**
  - Click on Test Connection to verify if the connection is established. On successful connection, the following message will be shown. See the figure below.



**Automature Plugin**

Zermatt Url  ?

User Name  ?

Password  ?

Successfully connected to server

## Create Job

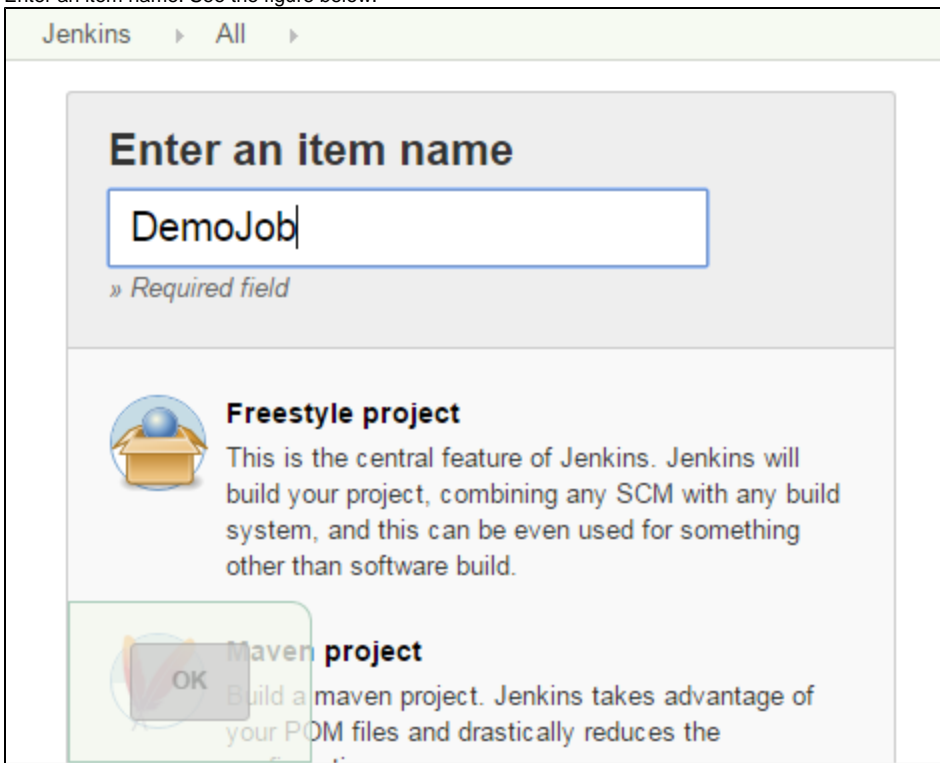
After successfully connecting to Zermatt server, create a job and configure Automature Jenkins Plugin for reporting.

### Steps Involved:

- Click on **New Item**. See the figure below.



- Enter an item name. See the figure below.



- Select **Freestyle project**. Click OK. See the figure below.

**Enter an item name**

DemoJob

» Required field

**Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the

Job created. For example:

**DemoJob** created as shown in the figure below.

Jenkins > DemoJob >

**General** Source Code Management Build Triggers

Build Environment Build Post-build Actions

Project name DemoJob

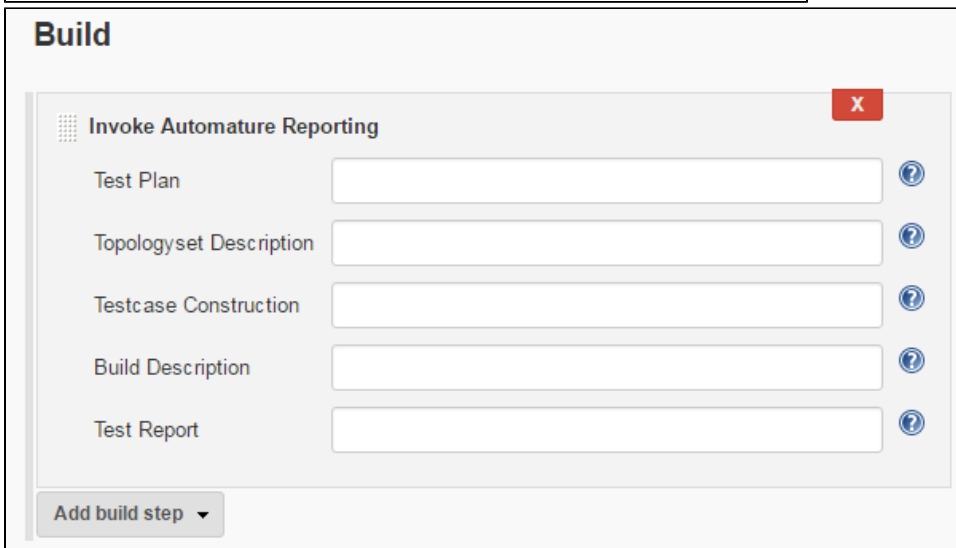
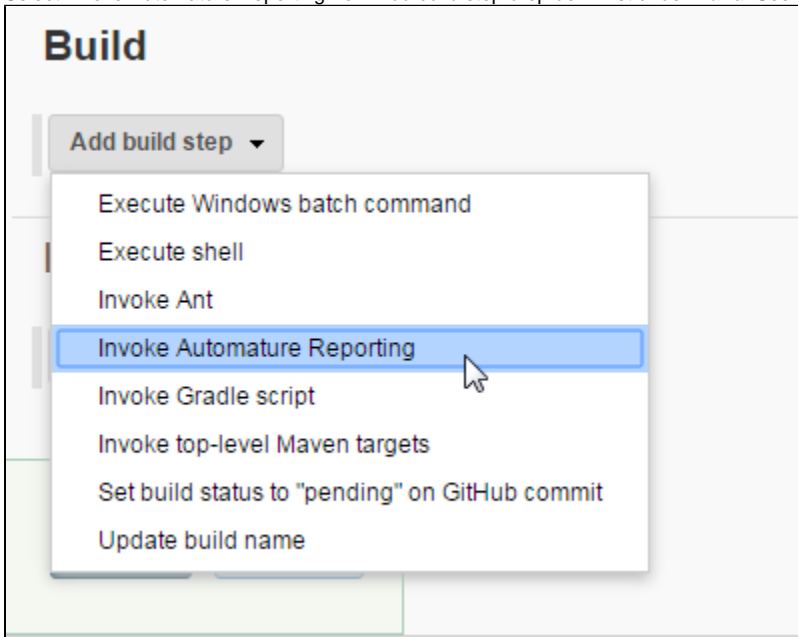
Description

## Configure Job

### Test Result Reporting

Steps Involved:

- Select Invoke Automature Reporting from Add build step drop down list under **Build**. See the figure below.

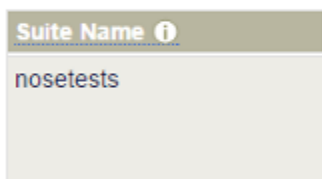


- **Test Plan**
  - Provide a Test Plan name where a test cycle will be created and test cases are reported.
    - The value of this field must contain a fully qualified Test Plan name preceded by the Product name separated by a dot(.).
      - For example, if product name is **DemoProduct** and test plan name is **DemoTestPlan** then the value of this field should be: **DemoProduct.DemoTestPlan**.
    - The product and testplan under consideration must exist in Zermatt. Create a new product and testplan if it does not exist.
- **Topologyset Description**
  - Provide Topologyset Description
    - If TopologySet Description is omitted, then a default TopologySet is assigned for the given Test Plan.
    - The topology (machine participating in the test execution) must exist in Zermatt. Create a new topology if it does not exist.
    - The topology must be associated with an existing topology set.
- **Testcase Naming**
  - Provide a format for the testcase identifier in Zermatt.
    - You may use one of the following formats viz.
      - @classname.@name
      - @name
      - @name\_jenkins
      - @classname\_jenkins
    - For example if in XML report, classname="democlassname" and name="demoname", the test case identifier constructed in Zermatt for @classname.@name, @name and @name\_jenkins is democlassname.demoname, demoname, demoname\_jenkins and democlassname\_jenkins respectively.
      - If the field is left blank then the Testcase identifier is the value of classname attribute.
- **Build Description**

- Provide a Build Description for product under consideration.
  - If Build Description is omitted then a default build is assigned.
- **Test Report**
  - The file must exist in the project's workspace.
  - If the file name is TestReports.xml then the value of the field is the file name itself.
  - If the file is located in a subdirectory called dir (at workspace) then the value of the field is dir/TestReports.xml.
  - The test report file, an xml file, must be in the following format as illustrated by the figure below

```
<?xml version="1.0" encoding="UTF-8"?>
<testsuite name="nosetests" tests="10" errors="0" failures="0" skip="0">
<testcase classname="test_parse.TestParse1" name="test_parse_modsec_correct_full_file" time="0.008"></testcase>
<testcase classname="test_parse.TestParse2" name="test_parse_modsec_default_values" time="0.002"></testcase>
<testcase classname="test_parse.TestParse3" name="test_parse_modsec_wrong_file" time="0.002"></testcase>
<testcase classname="test_settings.TestSettings" name="test_parse_args_verify_custom" time="0.007"></testcase>
</testsuite>
```

- The Suite Name in Zermatt will be the value of **name** attribute in the **<testsuite>** tag of test report. See the figure below.



- The Test Case identifier and Description of a test case in Zermatt are constructed from the values of **classname** and **name** attributes respectively in a **<testcase>** tag of the **xml** report. See the figure below.

Test Case	Action	Description
test_parse.TestParse3	[Icons: Refresh, Edit, Delete, List, Print]	test_parse_modsec_wrong_file
test_settings.TestSettings	[Icons: Refresh, Edit, Delete, List, Print]	test_parse_args_verify_custom
test_parse.TestParse1	[Icons: Refresh, Edit, Delete, List, Print]	test_parse_modsec_correct_full_file
test_parse.TestParse2	[Icons: Refresh, Edit, Delete, List, Print]	test_parse_modsec_default_values

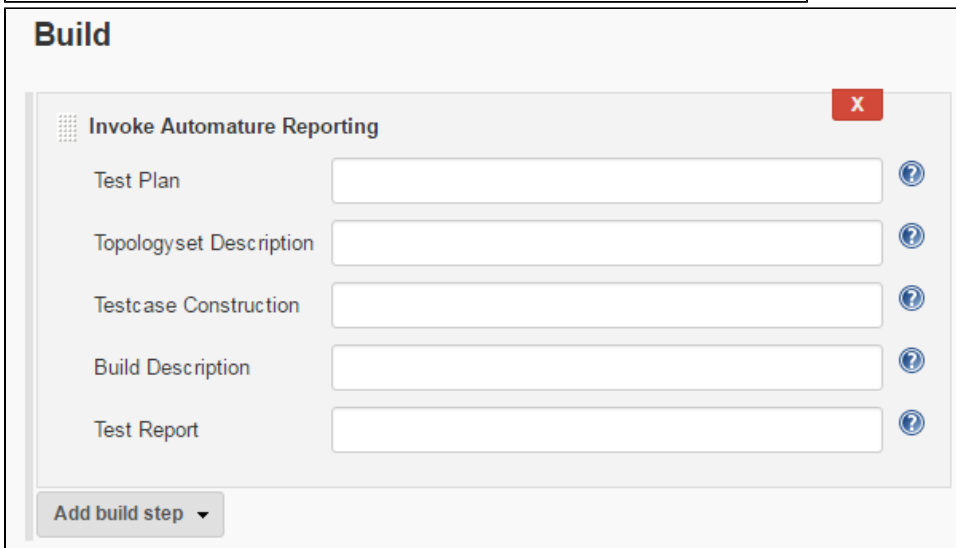
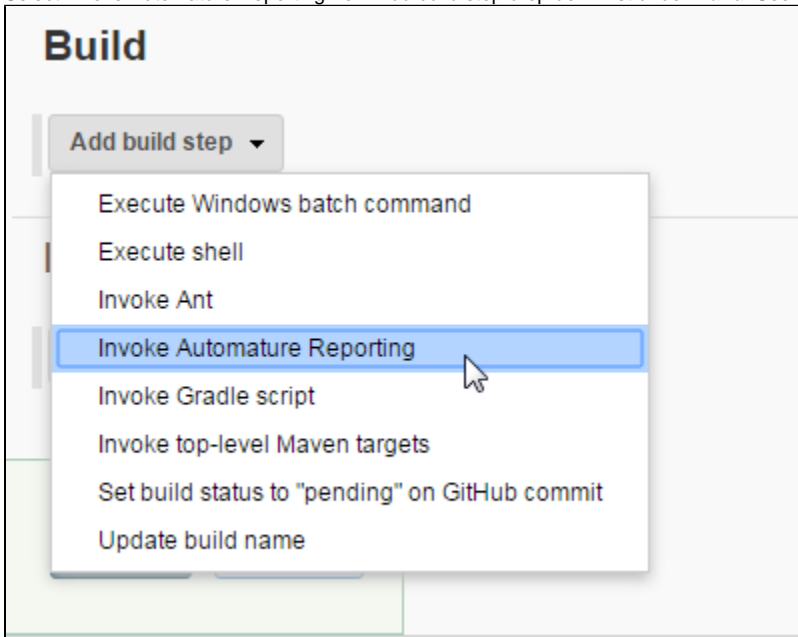
- **Save configuration**
  - Click on **Apply** and **Save** to save configuration.

## Test Coverage Reporting

This Plugin provides the ability to connect to Zermatt server and upload test coverage results from a XML coverage report file.

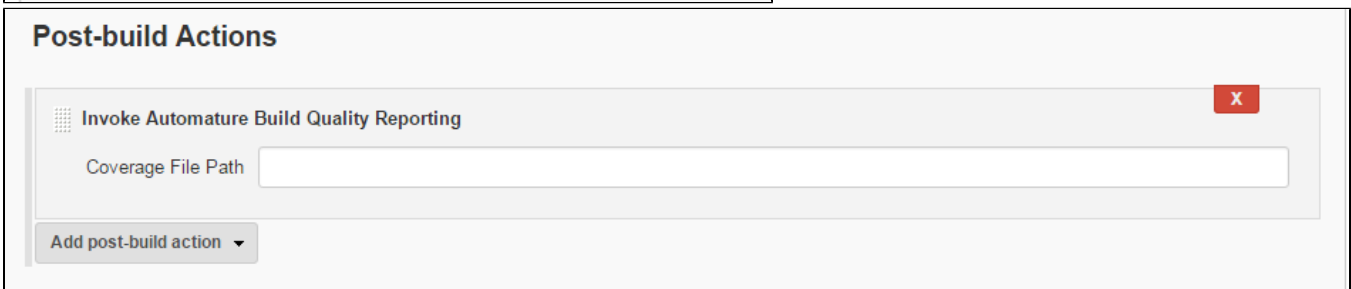
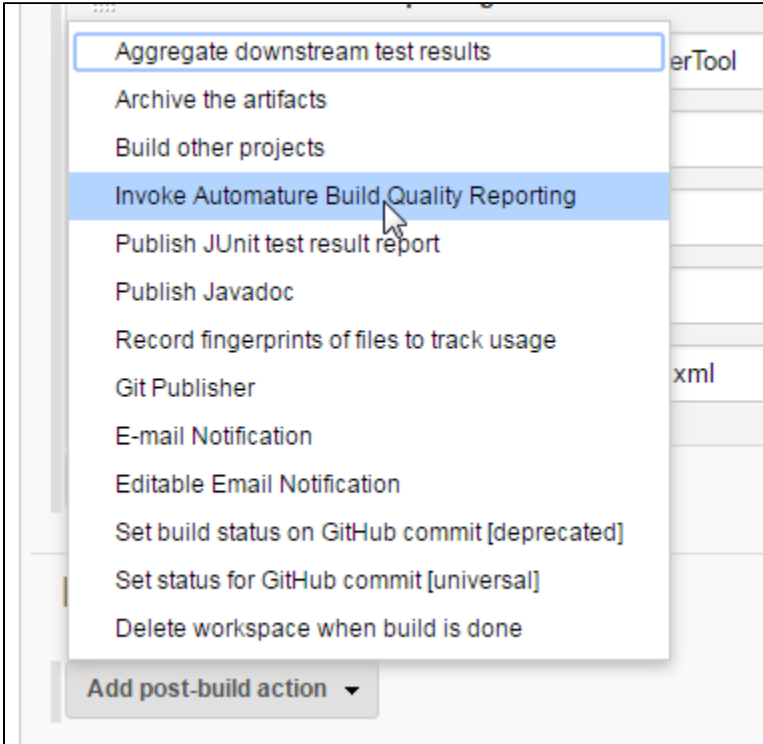
### Steps Involved:

- Select Invoke Automature Reporting from Add build step drop down list under **Build**. See the figure below.



- **Test Plan**
  - Provide a Test Plan name where a test cycle will be created and test cases are reported.
    - The value of this field must contain a fully qualified Test Plan name preceded by the Product name separated by a dot(.).
      - For example, if product name is **DemoProduct** and test plan name is **DemoTestPlan** then the value of this field should be: **DemoProduct.DemoTestPlan**.
    - The product and testplan under consideration must exist in Zermatt. Create a new product and testplan if it does not exist.
- **Topologyset Description**
  - Provide Topologyset Description
    - If TopologySet Description is omitted, then a default TopologySet is assigned for the given Test Plan.
    - The topology (machine participating in the test execution) must exist in Zermatt. Create a new topology if it does not exist.
    - The topology must be associated with an existing topology set.
- **Testcase Naming**
  - Provide a format for the testcase identifier in Zermatt.
    - You may use one of the following formats viz.
      - @classname.@name
      - @name
      - @name\_jenkins
      - @classname\_jenkins
    - For example if in XML report, classname="democlassname" and name="demoname", the test case identifier constructed in Zermatt for @classname.@name, @name and @name\_jenkins is democlassname.demoname, demoname, demoname\_jenkins and democlassname\_jenkins respectively.
      - If the field is left blank then the Testcase identifier is the value of classname attribute.
- **Build Description**

- Provide a Build Description for product under consideration.
  - If Build Description is omitted then a default build is assigned.
- **Test Report**
  - This field is optional.
    - If the field is left blank then a testcycle is created, no test results are reported, but coverage data is reported in the same testcycle.
    - If test report file path is provided, then a testcycle is created, test results are reported and coverage data is reported in the same testcycle.
      - The file must exist in the project's workspace.
      - If the file name is TestReports.xml then the value of the field is the file name itself.
      - If the file is located in a subdirectory called dir (at workspace) then the value of the field is dir/TestReports.xml.
- Select Invoke Automature Build Quality Reporting from Add post-build actions drop down list under Post-build Actions as illustrated below.



- **Coverage File Path**
  - The file must exist in the project's workspace.
  - If the file name is coverage.xml then the value of the field is the file name itself.
  - If the file is located in a subdirectory called dir (at workspace) then the value of the field is dir/coverage.xml.
  - The coverage data is reported to a test cycle in Zermatt as illustrated below.

ID	Cycle Description	Actions	Modification Date	Release	Sprint	Build	Topologysets	Failure Percentage	Execution Details	Code Coverage
3147	37_16-12-02 12:45		2016-12-02 12:44:54	KB2	KB2_2016_02	default_build	1		Total : 23 QA2 Test Area (Actual: 680 Failed: 10 Passed: 670 Aborted: 0)	INSTRUCTION : 0.18% BRANCH : 52.45% LINE : 66.36% COMPLEXITY : 1.23% METHOD : 0.20% CLASS : 20.67%

Testcycle created by plugin

Code Coverage Report

- **Save configuration**
  - Click on **Apply** and **Save** to save configuration.

## Zermatt Prerequisites

Below is a list of things you will need to configure in Zermatt before reporting your test execution results using the Automature Jenkins Plugin.

- **Create new user**
  - Existing users may skip this step.
- **Create new product**
  - Use an existing product or create a new product.
- **Add a machine**
  - Use an existing machine or configure a new machine.
    - A machine can be either a physical machine or a virtual machine. Later, this machine may be referenced as a part of a topology for a test session.
- **Add a role**
  - We normally use more than one machine in conducting a specific test. Each machine participates in a specific role that is assigned to it, for the purpose of a test.
- **Add a new topology for the role**
  - A Topology describes a single machine participating in the execution of a Test Case. The machine is described in terms of various significant attributes, such as its role, its operating system, the software installed on it, etc.
- **Add a new topology set**
  - A Topology Set describes a collection of machines required to execute a Test Case. Each machine may participate in a different role (e.g. client, server, database, Active Directory, proxy server, etc...). A test case may typically be executed on more than one topology set.
- **Associate topology to topology set**
- **Create test plan**
  - A test Plan identifies the following entities, viz.
    - The release and the project phase for which the plan exists.
    - The complete list of test suites (and specific test cases, contained in them) that should be executed during the course of the sprint.
    - The complete list of topology sets, where the test cases must be executed during the sprint.
- **Associate topology set to test plan**

The Automature Jenkins Plugin requires integration with other Automature products viz. Spacetime and Zermatt . To download Automature products, please visit [www.automature.com](http://www.automature.com).

## Change Log

### Version 1.3.0 (Mar 03, 2017)

- First release, split off from the core.