

Spirent APT Report Data Model and Report Template Customization

All of the company names and/or brand names and/or product names referred to in this document the name “Spirent” and its logo device are, or may be, either registered trademarks or trademarks of Spirent plc and its subsidiaries, pending registration in accordance with relevant national laws. All other registered trademarks or trademarks are the property of their respective owners

Contents

Overview	2
Report Data Model	2
Complete structure of the data model	4
Report Templates Customization.....	8
Highlight Execution Steps	8
Highlight Execution Issues	10
Set visibility of report sections	11
Additional Customization Options.....	13
Horizontal scroll bar.....	13
Change highlight color.....	13
Contacting Spirent.....	14

Overview

Velocity and iTest provides a custom template, which may be downloaded, modified as required and uploaded to generate your custom execution reports. The document **Spirent APT Report Data Model** describes the data model of the report template and provides examples of how to customize the template.

Note: Use the information provided in this document in conjunction with the sample report template provided (in Velocity and iTest).

Report Data Model

The report data model is a tree-like structure which contains the information necessary to generate a report. To refer to a model within a template, use the word `model`.

The data model has the following upper-level nodes:

report (hash):	contains general information about the execution report
steps (iterator):	Iterator for all steps of the execution
issues (iterator):	Iterator for all issues of the execution
attachments (iterator):	Iterator for all attachments of the execution
highlighting (hash):	a hash table that returns the highlight of the step response by its identifier
images (list):	a list of identifiers of images Which can be used in the report
headlineText (string):	report header text
velocityBaseUrl (string):	Velocity url which is used to load pictures and to open the topology preview
userToken (string):	
defaultImage (string):	image ID that should be used in place of the company logo, if it is not specified

The parentheses above indicate the variables type. The list of operations available differs depending on the type of the variable. Below is a brief description of the types:

Hash: stores other variables under a unique lookup name. The 'hash' data type implements the Freemarker interface TemplateHashModel. It contains two methods:

- `get(String key)`: returns the subvariable of the given name. Example of using in the default template: `model.report.testName`, where the `model` and `report` are hashes.
- `isEmpty()`: indicates whether the hash has zero subvariable

Sequences: store other variables in an ordered sequence (starting from 0)

- Numerical index can be used to retrieve the the stored variables
- Iterator: allows to sequentially iterate through the elements that are contained within the collection not using an index

Can be used inside of the loop, example of using in the default template:

```
<#list model.report.requirements as requirement>
  <@renderFormRow label=requirement.name field=requirement.value/>
</#list>
```

- list: provides the same capabilities as the iterator, but allows you to get the item by index
- string: string variable that can be inserted into the report. Examples of using in the default template:
 - To insert a value, use the following construction `${...}`, example of usage:

```
<div class="b-composite-title__item">
  <div class="b-text b-text_middle-title">${model.report.testName}</div>
</div>
```

- To verify that the value is not zero and not null,

you can use the following construction `...?has_content`, example of usage:

```
<#if model.headlineText?has_content>
  <div class="b-composite-title__item">
    <div class="b-text b-text_title">${model.headlineText}</div>
  </div>
</#if>
```

Complete structure of the data model

Below is the structure of the data model, If the node does not have a type in parentheses, then this is the string.

- Report(**hash**):

Fields	Description
testName	Name of the Automation Asset, e.g., super_X_steps_1.fttc
testPath	Location of the Automation Asset. E.g., main/exec-cs1b_large-9/test_cases/under-review/super_X_steps_1.fttc
owner	Name of the user executing the test. E.g., M Smith.
reportId	Report ID generated by Velocity. Example: 12f450ad-262e-434e-af0a-899c735c28a1
tags	Stored in the Velocity repository, supports %file% substitution for each tag separately.
detailLevel	Report detail level selected on the Automation Asset Details page before running the test. E.g.: Include all execution messages (and steps that have an error)
topologyId	Id of the topology used in the test execution.
startTime	The time when the execution started, e.g., Jun 9, 3:02 PM
endTime	The time when the execution ended, e.g., Jun 9, 4:02 PM
duration	The time taken to complete the test execution. E.g., 01:00:00.396
result	Result of Automation Asset execution. Pass, fail, and indeterminate, Abort, Cancel, Error. See Executions Report page (Reports > Executions), in the Online Help.
status	Execution status. E.g., Not Begun, Dispatching, In Progress, Completed, Start Failed, Aborted, and Agent Not Responding. See Executions Report page (Reports > Executions), in the Online Help.
failureReason	Reason for failure, if failed. See Executions Report page (Reports > Executions), in the Online Help.
agentHost	The host name where the agent is installed. E.g., sjc-742rv12.ad.spirentcom.com
agentName	Name of the agent that executed the test. E.g., AgentDebug
parameterFilePath	Location of the parameter file used in the test. E.g., main/exec-cs1b_large-95-d1/parameters/Landslide-GGSNNodal-Gn-GTP-Mslsdn.ffpt
topologyName	Name of the topology used in the test. E.g., TestGGSNNodal
totalItems	Indicates the total number report items. E.g., 1
totalIssues	Indicates the number of issues. E.g., 0
totalPass	Indicates the total number pass/OK messages. E.g., 2
totalFail	Indicates the total number failed Messages. E.g., 0

Fields	Description
totalError	Indicates the total number error messages. E.g., 1
totalWarning	Indicates the total number warning messages. E.g., 1
totalInfo	Indicates the total number information Messages. E.g., 5
parameters(list)	Indicates the parameters used to run the test. The parameters may be Automation Asset built-in parameters, parameters defined before running the test, or parameters contained within the selected parameter file.
	<ul style="list-style-type: none"> • Index: The string that reflects the position of the parameter in the parameters tree. E.g., the first parameter will have an index "1" and its first child parameter will have an index "1.1"
	<ul style="list-style-type: none"> • Name: Parameter name, which is used to refer the parameter in the test. E.g., "firmwareRev".
	<ul style="list-style-type: none"> • value: Parameter value. E.g., "3.141".
requirements(list)	Indicates the Agent capabilities requirements name value pair defined in the test.
	<ul style="list-style-type: none"> • Name: The name of the agent capability to which this requirement applies. E.g, "os.type".
	<ul style="list-style-type: none"> • Value: Required value of the agent capability. E.g, "win32".

- **steps(iterator):** Indicates the steps as defined for execution.

A step is a single row defined in a, iTest Test Case that often represents a command. A step may be defined to also perform a program control operation like a **for** loop or **comment**, or it can perform an **open** or **close** action to start or end a session.

Fields	Description
stepId	Step ID generated by the agent.
action	Action of the step. E.g., "eval".
command	Command of the step. E.g., "puts "hello""
startOffset	The elapsed time from the start of the test execution to the start of this step execution
duration	Step execution duration.

threadId	The thread ID in which the step was executed
response	
Documentation	Label, Tag, Comment: A description of the step properties for documentation purposes.
postProcessing(list)	A list of actions that are performed after the execution of the step.
	<ul style="list-style-type: none"> • index: The index reflecting the position of the action in the action tree, e. g. the first action will have an index "1" and its first child action will have an index "1.1"
	<ul style="list-style-type: none"> • Action: The name of the action, e.g. "analyze"
	<ul style="list-style-type: none"> • Description: The description of the action, e.g. "regex hello: assert \$value == "hello""
artifactLinks(list)	A list of artifacts used in the test execution.
	<ul style="list-style-type: none"> • description: The description of the artifacts generated during test execution.
	<ul style="list-style-type: none"> • link: Link to the artifact stored on an external file system.

- issues(**Iterator**): Indicates any issues related to the steps executed.

Fields	Description
stepId	Step ID generated by the agent.
originator	The map or line in the response in which the agent has detected the issue.
severity	Severity level of the execution message, it can be one of the following values: "PASS", "INFORMATION", "WARNING", "ERROR".
message	Description of the issue, including execution messages that are generated by analysis rules.

- attachments(**Iterator**): Indicates charts generated during test case execution

descriptor	Descriptor of the attachment. At the moment, this descriptor is not used directly in the template, but is used by the report generator to get the content of the attachment.
------------	--

	<ul style="list-style-type: none"> reportId: The ID of the report with which this data is associated.
	<ul style="list-style-type: none"> stepId: The ID of the executed step with which this data is associated. E.g., "1.2.1"
	<ul style="list-style-type: none"> meta: Arbitrary type of file. E.g., "screenshot".
	<ul style="list-style-type: none"> Name: Name of the file. E.g., "chart.png".
contentBase64	Base64 content of the attachment.

- highlighting(**hash**): Indicates highlighting blocks for response by a given stepId. These blocks are used to render the highlighting in the step responses. Highlighting block structure:

line	The line number on which the highlight begins
lineCount	Number of rows covered by the highlight
startColumn	The column number on which the highlight begins
endColumn	The column number on which the highlight ends
endOnOtherLine	Flag that indicates that the highlight flows to the next row
severity	Number indicating the severity level of the issue: 1 - information, 2 - pass, 3 - warning, 4 - error. Depending on the severity, the color of the block will differ: 1 - blue, 2 - green, 3 - orange, 4 - red.

- images(**list**)

modelId	ID of the image in data model, used to access the data model from the template
imageId	ID of the image that will be used in the template

- headlineText: Indicates the title text
- velocityBaseUrl: Identifies the Velocity URL which is used to load resources and display topology
- userToken: Identifies the token that is used to load resources from the Velocity. E.g., is used to render default company logo.
- defaultImage: Indicates the default image used as the logo when printing execution report with the default template.

Report Templates Customization

The sections below provide you with working example of how to customize your template:

- [Highlight Execution Steps](#)
- [Highlight Execution Issues](#)
- [Set Visibility of report sections](#)
- [Additional Customization Options](#)
 - [Horizontal scroll bar](#)
 - [Change highlight color](#)

Highlight Execution Steps

You may specify which steps and execution issues that you wish to highlight in the `steps` field of the `HIGHLIGHT_MAP` variable at the beginning of the default template.

To highlight an element of the Steps node, set the following fields:

- **field:** the name of one of the string fields in the step data model. The following fields can be specified:
 - stepId
 - action
 - command
 - threadId
 - sessionId
 - sessionType
 - startOffset
 - duration
- **pattern:** if the step field is equal to this value, the step will be highlighted
- **color:** highlight color

Example of highlighting the Steps node by the type of action:

```
HIGHLIGHT_MAP={
  "steps": [
    {
      "field": "action",
      "pattern": "call",
      "color": "#dcecf9"
    },
    {
      "field": "action",
      "pattern": "open",
      "color": "#dbefdf"
    },
    {
      "field": "action",
      "pattern": "command",
      "color": "#fbfbe7"
    }
  ],
  "issues": []
}
```


As a result, the steps will be highlighted as follows:

Steps

Step #	Action	Start Time	Duration																														
	Action: call Command: main POST PROCESSING	0.228	1.489																														
	<table border="1"> <thead> <tr> <th>#</th> <th>Action</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>event</td> <td>OnProcedureEnter</td> </tr> <tr> <td>2</td> <td>event</td> <td>OnThreadEnter</td> </tr> <tr> <td>3</td> <td>event</td> <td>OnExecutionStarted</td> </tr> <tr> <td>3.1</td> <td>action</td> <td>DeclareExecutionIssue Information:Execution started</td> </tr> <tr> <td>4</td> <td>event</td> <td>OnProcedureExit</td> </tr> <tr> <td>5</td> <td>event</td> <td>OnThreadExit</td> </tr> <tr> <td>6</td> <td>event</td> <td>OnPreExecutionCompleted</td> </tr> <tr> <td>7</td> <td>event</td> <td>OnExecutionCompleted</td> </tr> <tr> <td>7.1</td> <td>action</td> <td>DeclareExecutionIssue Information:Execution completed (1s)</td> </tr> </tbody> </table>	#	Action	Description	1	event	OnProcedureEnter	2	event	OnThreadEnter	3	event	OnExecutionStarted	3.1	action	DeclareExecutionIssue Information:Execution started	4	event	OnProcedureExit	5	event	OnThreadExit	6	event	OnPreExecutionCompleted	7	event	OnExecutionCompleted	7.1	action	DeclareExecutionIssue Information:Execution completed (1s)		
#	Action	Description																															
1	event	OnProcedureEnter																															
2	event	OnThreadEnter																															
3	event	OnExecutionStarted																															
3.1	action	DeclareExecutionIssue Information:Execution started																															
4	event	OnProcedureExit																															
5	event	OnThreadExit																															
6	event	OnPreExecutionCompleted																															
7	event	OnExecutionCompleted																															
7.1	action	DeclareExecutionIssue Information:Execution completed (1s)																															
1	Action: open Command: project://wireshark_oom/session_profiles/Wireshark.ffsp RESPONSE	1.271	0.303																														
	<pre>Spirent Communications Wireshark command interpreter. Copyright (c) 2005 - 2017, The Spirent Communications</pre>																																
2	Action: command Command: capture load project://wireshark_oom/pcap/15000.pcap POST PROCESSING	1.593	0.022																														
	<table border="1"> <thead> <tr> <th>#</th> <th>Action</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>event</td> <td>OnInvalidStepPropertyType</td> </tr> <tr> <td>1.1</td> <td>action</td> <td>DeclareExecutionIssue Error:Application properties associated with the step are of a type that is invalid for this step: OpenStepPropertyGroup.</td> </tr> <tr> <td>1.2</td> <td>action</td> <td>FailTest</td> </tr> <tr> <td>1.2.1</td> <td>event</td> <td>OnTestResultChange</td> </tr> <tr> <td>1.2.2</td> <td>event</td> <td>OnFailTestAction</td> </tr> <tr> <td>1.3</td> <td>action</td> <td>AbortStep</td> </tr> <tr> <td>2</td> <td>event</td> <td>OnFailedTestResult</td> </tr> <tr> <td>2.1</td> <td>action</td> <td>DeclareExecutionIssue Error:Test case wireshark_oom has failed.</td> </tr> </tbody> </table>	#	Action	Description	1	event	OnInvalidStepPropertyType	1.1	action	DeclareExecutionIssue Error:Application properties associated with the step are of a type that is invalid for this step: OpenStepPropertyGroup.	1.2	action	FailTest	1.2.1	event	OnTestResultChange	1.2.2	event	OnFailTestAction	1.3	action	AbortStep	2	event	OnFailedTestResult	2.1	action	DeclareExecutionIssue Error:Test case wireshark_oom has failed.					
#	Action	Description																															
1	event	OnInvalidStepPropertyType																															
1.1	action	DeclareExecutionIssue Error:Application properties associated with the step are of a type that is invalid for this step: OpenStepPropertyGroup.																															
1.2	action	FailTest																															
1.2.1	event	OnTestResultChange																															
1.2.2	event	OnFailTestAction																															
1.3	action	AbortStep																															
2	event	OnFailedTestResult																															
2.1	action	DeclareExecutionIssue Error:Test case wireshark_oom has failed.																															

Highlight Execution Issues

Highlighting an execution issue is the same as for the steps. Ensure that you need to specify the fields relevant for the execution issues:

- stepId
- originator
- severity
- message

Example of highlighting messages by originator:

```
HIGHLIGHT_MAP={
  "steps": [],
  "issues": [
    {
      "field": "originator",
      "pattern": "execution",
      "color": "#dcecf9"
    },
    {
      "field": "originator",
      "pattern": "comment",
      "color": "#dbefdf"
    },
    {
      "field": "originator",
      "pattern": "message",
      "color": "#fbfbe7"
    }
  ]
}
```

As a result, the execution messages will be highlighted as follows:

Execution Messages

Step #	Originator	Message
	execution	Execution started
1	comment	parse the topology for vlan structures
1.1	exec.eval	Unable to evaluate "set vlanObjIdList [tbml query //tbml/body/resources/resource\[@type='vlan'\]@id]": Cannot find the topology file
1.1	declareExecutionIssue.message	Unable to substitute " vlanObjIdList: \$vlanObjIdList" : Variable "vlanObjIdList" does not exist
1.1	message	vlanObjIdList: \$vlanObjIdList
1.2	exec.if	Unable to evaluate "[length \$vlanObjIdList] > 0" : Variable "vlanObjIdList" does not exist
1.3	exec.eval	Unable to evaluate "set deviceNames [tbml query //tbml/body/resources/resource/property\[@name='name'\]]": Cannot find the topology file
1.3	declareExecutionIssue.message	Unable to substitute " Device list: \$deviceNames" : Variable "deviceNames" does not exist
1.3	message	Device list: \$deviceNames
2	comment	show the parameter
2	message	param1 value is hello world!
2.1	analysis	Passed
1.3	execution	Test case helloworld has failed.
	execution	Execution completed (0s)

Set visibility of report sections

You may configure the sections of the report that should be visible.

By default, all sections are visible and is configured as arguments of the variable `VISIBILITY_OF_REPORT_SECTIONS` at the beginning of the template.

The following is the declaration of the VISIBILITY_OF_REPORT_SECTIONS variable in the default template:

```
VISIBILITY_OF_REPORT_SECTIONS = {
  "information": true,
  "statistics": true,
  "parameters": true,
  "requirements": true,
  "executionMessages": {
    "ERROR": true,
    "WARNING": true,
    "PASS": true,
    "INFORMATION": true
  },
  "steps": true
}
```

Tip: To hide one of the sections, replace the value with `false`.

Example of filtering execution messages by severity (removes warning and information messages):

```
VISIBILITY_OF_REPORT_SECTIONS = {
  "information": true,
  "statistics": true,
  "parameters": true,
  "requirements": true,
  "executionMessages": {
    "ERROR": true,
    "WARNING": false,
    "PASS": true,
    "INFORMATION": false
  },
  "steps": true
}
```

The screenshots below show how messages look without filters and with filters:

Execution Messages

Step #	Originator	Message
	execution	Execution started
1	analysis	Response contains "response"
2	analysis	Response contains "One"
2	analysis	Response contains "One,"
2	analysis	Response contains "One,"
2	analysis	Response contains "two"
2	analysis	Response contains "three"
2	analysis	Response contains "three!"
2	analysis	Response contains "Four, five, six, seven"
2	analysis	Response contains "Four"
2	analysis	Response contains "nine, ten"
2	analysis	Response contains "twelve!"
2	analysis	Response contains "Bye."
2	analysis	AAA
2	analysis	fff
2	analysis	Response contains "fifteen"
2	analysis	Response contains "\$"
2	analysis	Matched value "test", which is not equal to "1"
2	analysis	Response contains " "
2	analysis	Response contains "test777"
3	analysis	Response contains "One"
3	analysis	Response contains "One,"
3	analysis	Response contains "One,"
3	analysis	Response contains "two"
3	analysis	Response contains "three"
3	analysis	Response contains "three!"
3	analysis	Response contains "Four, five, six, seven"
3	analysis	Response contains "Four"
3	analysis	Response contains "nine, ten"
3	analysis	Response contains "twelve!"
3	analysis	Response contains "Bye."
3	analysis	AAA
3	analysis	fff
3	analysis	Response contains "fifteen"
3	analysis	Response contains "\$"
3	analysis	Matched value "test", which is not equal to "1"
3	analysis	Response contains " "
3	analysis	Response contains "test666"
3	execution	Test case response_highlighting has failed.
	execution	Execution completed (1s)

Execution Messages

Step #	Originator	Message
1	analysis	Response contains "response"
2	analysis	Response contains "One"
2	analysis	Response contains "One,"
2	analysis	Response contains "two"
2	analysis	Response contains "Four, five, six, seven"
2	analysis	Response contains "Four"
2	analysis	Response contains "Bye."
2	analysis	AAA
2	analysis	Response contains "\$"
2	analysis	Matched value "test", which is not equal to "1"
2	analysis	Response contains "test777"
3	analysis	Response contains "One"
3	analysis	Response contains "One,"
3	analysis	Response contains "two"
3	analysis	Response contains "Four, five, six, seven"
3	analysis	Response contains "Four"
3	analysis	Response contains "Bye."
3	analysis	AAA
3	analysis	Response contains "\$"
3	analysis	Matched value "test", which is not equal to "1"
3	analysis	Response contains "test666"
3	execution	Test case response_highlighting has failed.

Additional Customization Options

Horizontal scroll bar

You may enable or disable horizontal scroll bar to view wide step responses by changing variable `ENABLE_SCROLLING_IN_STEP_RESPONSE`. Horizontal scrolling is enabled by default.

```

3.1 Action: eval                                0.479      0.122
    Command: puts [velocity token]
    RESPONSE
    eyJwYWNRZXQioiAielwidXNlc19uYW11XCI6IFwidGVzdGFkbWluXCIsIFwiZG9tYWluXCI6IFwibGRhcFwiLCBcI
  
```

Change highlight color

You may change the highlight color, for example, of the step response by changing variables: RESPONSE_INFO_COLOR, RESPONSE_PASS_COLOR, etc.

```

3.1 Action: eval                                0.479      0.122
    Command: puts [velocity token]
    RESPONSE
    eyJwYWNRZXQioiAielwidXNlc19uYW11XCI6IFwidGVzdGFkbWluXCIsIFwiZG9tYWluXCI6IFwibGRhcFwiLCB
    cImlkXCI6IFwidGVzdGFkbWluXCIsIFwiZlhwXJhdG1vb1wiOiaXNTI3ODU2MTI2LCBcInN0YXJ0X3RpbWVcIj
    ogMTQ5NjMxOTgyNn0iLCAic2lnbmF0dXJlIjogIjQ2ZjE2NmNhNTM0YjA0YWNkM2UzMzU3jQ3ZDRkZ
    TVjYmY4NTJmMTNlNTE0ODIwNTA2MWEyYjA2YTQ4NTVmY2FmY2NlNWQyODFmYTg0ZjU3M2QzM2MzYzglZmJkMzc1
    NDI4YzY3MmY3NmRkMTIyZjVlZmZlMjNiZmRmOWNhMTAzNWZOTVLODU1MTg0OTNmYzc0OThiZjdmNWExODI4ZTN
    mZGYlZTBmYmFjNzBmZu0NjM2OTA5N2U0YmY5MjhiZTkwnNhOTg0ZmYzZGIxOWUxNjE1NDAXNzElMTQ5NzFlZD
    Q3ZWRjNDRlN2UyYTM1ZDAzMWZmMjc1MTMifQ==
  
```

Contacting Spirent

How to Contact Us

To obtain technical support for any Spirent Communications product, please contact our Support Services department using any of the following methods:

Americas

Web: <https://support.spirent.com>

E-mail: support@spirent.com

Toll Free: North America

+1 800-SPIRENT

(+1 800-774-7368)

Spirent Federal (U.S.)

Web: <https://support.spirent.com>

E-mail: help@spirentfederal.com

Phone: +1 801-785-1275

Europe, Middle East, Africa (EMEA)

Web: <https://support.spirent.com>

E-mail: support@spirent.com

EMEA: +33 (1) 6137-2270

United Kingdom: +44 1803-546333

Toll Free: +1 818-676-2616

Asia Pacific

Web: <https://support.spirent.com>

E-mail: support@spirent.com

In China Mainland:

86 (400) 810-9529 (toll-free)

Out of China Mainland:

+86 10 8233-0033

India: 1800-419-2111

Information about Spirent Communications and its products and services can be found on the main company website at <https://www.spirent.com>.

Information about our global office locations can be found at [https://www.spirent.com/About Us/Locations](https://www.spirent.com/About_Us/Locations)

Page intentionally left blank