

DAVOS USER MANUAL		Pages (1/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

Davos User Manual (Davos V6.4.0)

Whilst all reasonable care has been taken to ensure that the details are true and not misleading at the time of publication, no liability whatsoever is assumed by Automature LLC, or any supplier of Automature LLC, with respect to the accuracy or any use of the information provided herein.

Any license, delivery and support of software require entering into separate agreements with Automature LLC.

This document may contain confidential information and may not be modified or reproduced, in whole or in part, or transmitted in any form to any third party, without the written approval from Automature LLC.

Copyright © 2012 Automature LLC

All rights reserved.

DAVOS USER MANUAL		Pages (2/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

Revision History

Ver.	Date	Name	Description of Changes
1	14-09-2011	Sankha Sil	First Edition
2	06-01-2012	Debabrata Das	Revision
3	27-01-2012	Debabrata Das	Revision
4.2	15-06-2012	Chayan Hazra	Second Edition
4.4	19-06-2012	Chayan Hazra	Revised Edition
6.3	28-06-2013	Chayan Hazra	Revised Edition
6.4	30-07-2013	Chayan Hazra	Revised Edition
6.4.1	31-07-2013	Chayan Hazra	Revised Edition

DAVOS USER MANUAL		Pages (3/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

Contents

Introduction	6
1.1 Document Purpose	6
1.2 Intended Audience	6
Davos Interfaces	7
2.1 Session Management	7
2.1.1 session/create	7
2.1.2 session/destroy	8
2.1.3 /updateSession	9
2.1.4 /heartBeat	9
2.2 Getting Planning Data	10
2.2.1 getZermattConfig/value	10
2.2.2 role/read	11
2.2.4 topologyset/read	11
2.2.5 /getTestplanTopologies	12
2.2.6 /getTestCycleXML	13
2.3 Posting Planning Data	15
2.3.1 create/objects	15
2.3.2 release/write	16
2.3.4 sprint/write	17
2.3.5 build/write	18
2.3.6 testplan/write	19
2.3.7 testsuite/write	20
2.3.8 testcase/write	21
2.3.9 testcycle/write	22
2.3.10 variables/write	24
2.4 Posting execution Data	25
2.4.1 testexecutiondetails/write	25
2.4.2 performancemetric/initialize	27
2.4.3 performance/write	27
2.4.4 performancemetric/finalize	29
2.4.5 /setTestCycleTopologySetValues	29
2.5 Other helper calls	31
2.5.1 /updateMyMachine	31
2.5.2 /renameFile	31
2.5.3 /sendmail	33
2.5.4 /commitNyonJob	34
Test execution reporting without ZUG	35

DAVOS USER MANUAL		Pages (4/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

DAVOS USER MANUAL		Pages (5/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

1 Introduction

Davos is the Web Service Interface to Zermatt, Automature's QA Management tool. . Davos exposes an API that can be used to store test results in Zermatt and query planning data. Davos is used primarily by Automature's other products, viz. Zug and Nyon to interact with Zermatt.

Davos is also intended to expose Zermatt to other automation tools, that want to report results into Zermatt.

Davos is implemented using Ruby, and is based on the Sinatra web-services framework.

1.1 Document Purpose

This User Manual explains how to call and store test execution results through Davos into Zermatt. Its primary intent is to act as a reference guide for the exposed API (REST services). It is not intended to act as an indepth guide to integration.

1.2 Intended Audience

This manual is intended for tool integrators, who need to understand the REST API. The manual may also help other technical users, who want to look “under the covers” to gain insight into the inner workings of Automature products.

For normal users of Automature products, this manual adds only marginal value, and might be too detailed and incoherent.

DAVOS USER MANUAL		Pages (6/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

2 Davos Interfaces

2.1 Session Management

2.1.1 session/create

Creates a session and returns an id to be used by the client for making other requests

Returns session_id in xml format.

- **Pre-Conditions**

User must be present in my-sql database having permission over davos database

- **Summary**

URL: http://www.example.com/session/create

Format: XML

Authentication: true

HTTP Method(s): POST

³⁵₁₇ **Request**

HTTP POST Request Example

user=username&password=password

³⁵₁₇ **Response**

XML Example

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<session>
```

```
  <id>1234</id>
```

```
</session>
```

OR

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<davos>
```

```
  <error> You are not authorized to view this page! </error>
```

```
</davos>
```

DAVOS USER MANUAL		Pages (7/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

2.1.2 session/destroy

Destroys a session and removes all session related data from the server

³⁵₁₇ **Summary**

URL:http://www.example.com/session/destroy

Format: XML

Authentication: true

HTTP Method(s): POST

³⁵₁₇ **Request**

HTTP POST Request Example

session_id=1234

³⁵₁₇ **Response**

XML Example

```
<?xml version="1.0" encoding="utf-8" ?>
<session_Data>
  <id>Session Destroyed Successfully</id>
</session_Data>
```

OR

```
<?xml version="1.0" encoding="utf-8" ?>
<davos>
  <error> Session Timed Out. Please login again and continue.</error>
</davos>
```

DAVOS USER MANUAL		Pages (8/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

2.1.3 /updateSession

Keeps the session alive for a particular session id

³⁵/₁₇ *Summary*

URL:http://www.example.com/updateSession

Format: XML

Authentication: true

HTTP Method(s): POST

³⁵/₁₇ *Request*

HTTP POST Request Example

session_id=1234

2.1.4 /heartBeat

Keeps the session alive.

³⁵/₁₇ *Summary*

URL:http://www.example.com/heartBeat

Formats: Xml

Authentication: false

HTTP Method(s): GET

³⁵/₁₇ *Request*

HTTP POST Request Example

session_id=1234

³⁵/₁₇ *Response*

XML Example

DAVOS USER MANUAL		Pages (9/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<Session_Data>
```

```
<id>1234</id>
```

```
</Session_Data>
```

OR

```
<davos>
```

```
<error> Session Timed Out. Please login again and continue.</error>
```

```
</davos>
```

2.2 Getting Planning Data

2.2.1 getZermattConfig/value

Returns the configuration value of individual Zermatt config attribute in xml format.

- **Summary**

URL: <http://www.example.com/getZermattConfig/value>

Format: XML

Authentication: false

HTTP Method(s): GET

- **Request**

HTTP POST Request Example

name=Archivelocation&session_id=1234

- **Response**

XML Example

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<zermattConfig><value>\\192.168.0.2\share </value></zermattConfig>
```

OR

```
<zermattConfig><error> Key name missing! </error></zermattConfig>
```

OR

DAVOS USER MANUAL		Pages (10/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

*<zermattConfig><error> Key not found! Key Name = Archivelocation
</error></zermattConfig>*

2.2.2 role/read

Returns the role id for a given role name in xml format.

³⁵₁₇ *Summary*

URL:http://www.example.com/role/read

Format: XML

Authentication: false

HTTP Method(s): GET

³⁵₁₇ *Request*

HTTP POST Request Example:

role_name=some_role&session_id=1234

³⁵₁₇ *Response*

XML Example

<?xml version="1.0" encoding="utf-8" ?>

<role><id>1234</id></role>

OR

<role><error> Role name missing! </error></role>

OR

<role><error> Role not found! Role Name = some_role </error></role>

2.2.4 topologysset/read

Returns the topologysset id for a given topologysset name or validates a topologysset id.

³⁵₁₇ *Summary*

URL:http://www.example.com/topologysset/read

Format: XML

Authentication: false

DAVOS USER MANUAL		Pages (11/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

HTTP Method(s): Post

³⁵₁₇ **Request**

HTTP POST Request Example

topologysset_description=some_topologysset&session_id=1234

OR

topologysset_id=540&session_id=1234

³⁵₁₇ **Response**

XML Example

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<topologysset><id>1234</id></topologysset>
```

OR

```
<davos><error> Topologysset not found! Topologysset Description = some_topologysset
</error></davos>
```

OR

```
<davos><error> Topologysset not found! Topologysset Id = 540</error></davos>
```

OR

```
<davos><error> Topologysset description or id not defined!</error></davos>
```

2.2.5 /getTestplanTopologies

Returns a xml, containing a testcycle_id and machine configurations of the test plan.

- **Pre-Conditions**

The test plan must be associated with some testsuite and some topologysset. The topology role must match with the role of the testsuite.

³⁵₁₇ **Summary**

URL:http://www.example.com/variables/write

Format: XML

Authentication: false

HTTP Method(s): POST

³⁵₁₇ **Request**

HTTP POST Request Example

DAVOS USER MANUAL		Pages (12/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

testcycle_id=111&session_id=1234

³⁵₁₇ **Response**

XML Example

```
<?xml version="1.0" encoding="utf-8" ?>
<Root>
  <TESTCYCLE ID="111">
    <MACHINES>
      <MACHINE IP="192.168.0.111" NAME="LADAKH" PORT="4569"/>
    </MACHINES>
  </TESTCYCLE>
</Root>
```

OR

```
<davos><error> Test Cycle does not exist! Test Cycle Id = 111 </error></davos>
```

OR

```
<davos><error> No matching machines detected to launch NYON!</error></davos>
```

2.2.6 /getTestCycleXML

Returns a xml, containing a testcycle_id, build_id, testplan_id, list of testsuites and machine configurations as designed in the test plan.

- **Pre-Conditions**

The test plan must be associated with some testsuite and some topologyset. The topology role must match with the role of the testsuite.

ZermattPubURL must be defined in frameworkmetadata table.

³⁵₁₇ **Summary**

URL:http://www.example.com/getTestCycleXML

Format: XML

Authentication: false

HTTP Method(s): POST

³⁵₁₇ **Request**

HTTP POST Request Example

testcycle_id=544&session_id=1234

DAVOS USER MANUAL		Pages (13/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

35 17 **Response**

XML Example

```
<?xml version="1.0" encoding="utf-8" ?>
<ROOT>
<TESTCYCLE ID='544'>
<TESTPLAN ID='893' OPTIONS=' ' PRODUCT='JUPITER' PRODUCTID='111'>
<BUILD ID='0'>
<EXECUTION_DETAILS>
<EXECUTION TOPOLOGYSET_NAME='Local Testing' TOPOLOGYSET_ID='25'
TESTSUITE_ID='320' TESTSUITE_NAME='MVCV'
PATH='http://zermatt.manimitra.in/twiki/pub/ZERMATT/AttachTestSuiteFile/mvcv.xls'
TSDIGEST='qwttyhdgt56husujssiisw89'
ARCHIVEPATH='http://zermatt.manimitra.in/twiki/pub/ZERMATT/AttachTestSuiteDataZipF
ile/mvcv.xls' ARCHIVEDIGEST='567hrtyr784kj98jei93k20210ll' ROLE='Zug-
Client'><OPTIONS><OPTION NAME='TESTCASE'
VALUE='MVCV002_value1,MVCV003_value2,MVCV004_c'>
</OPTIONS>
</EXECUTION>
</EXECUTION_DETAILS>
</BUILD>
</TESTPLAN>
</TESTCYCLE>
</ROOT>
```

OR

```
<davos><error> Testcycle Id missing! </error></davos>
```

OR

```
<davos><error> No testsuite associated with this testplan! Test Plan Id = 893
</error></davos>
```

OR

```
<davos><error> No topology set associated with this testplan! Test Plan Id = 893
</error></davos>
```

OR

```
<davos><error> This machine does not belong to any of the topology sets selected! Machine
IP Address = 192.168.0.111 </error></davos>
```

OR

```
<davos><error> Test cycle not found! Test Cycle Id =544 </error></davos>
```

OR

```
<davos><error> ZermattPubURL not defined in ZERMATT. </error></davos>
```

OR

```
<davos><error> Error while downloading file!</error></davos>
```

DAVOS USER MANUAL		Pages (14/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

2.3 Posting Planning Data

2.3.1 create/objects

Creates different database objects in the order Release, Sprint, Test plan, only if not present. It takes a colon separated list of database objects in the order Product, Release, Sprint, Testplan.

³⁵/₁₇ **Pre-Condition**

Product must be present.

³⁵/₁₇ **Summary**

URL: <http://www.example.com//create/objects>

Formats: Xml

Authentication: false

HTTP Method(s): POST

³⁵/₁₇ **Request**

HTTP POST Request Example

[session_id=1234&build_path=dummmmy_product:dummy_release:dummy_sprint:dummy_test plan](#)

³⁵/₁₇ **Response**

XML Example

[<?xml version="1.0" encoding="utf-8" ?>](#)

[<testplan><id>34</id></testplan>](#)

OR

[<davos><error> Object names not defined! </error></davos>](#)

OR

[<davos><error> Incorrect number of object names defined! Object Names =dummmmy_product:dummy_release:dummy_sprint, Length =4 </error></davos>](#)

OR

[<davos><error> Product not found! Product Name = dummmmy_product</error></davos>](#)

DAVOS USER MANUAL		Pages (15/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

2.3.2 release/write

Creates release entry with the code_name under the product name specified. If a release already exists with the given name, then its id is returned.

Returns release_id in xml format.

By default the conceived date is today's date and released date is 1 year from the conceived date.

³⁵/₁₇ **Pre-Conditions**

Product must be present.

³⁵/₁₇ **Summary**

URL: <http://www.example.com/release/write>

Format: XML

Authentication: false

HTTP Method(s): POST

³⁵/₁₇ **Request**

HTTP POST Request Example

[session_id=1234&product_name=dummy_product&code_name=dummy_release](#)

³⁵/₁₇ **Response**

XML Example:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<release><id>34</id></release>
```

OR

```
<davos><error> Code name not defined! </error></davos>
```

OR

```
<davos><error> Product not found. Product Name = dummy_product</error></davos>
```

OR

```
<davos><error> Product name not defined! </error></davos>
```

DAVOS USER MANUAL		Pages (16/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

2.3.4 sprint/write

Creates sprint entry with the sprint_name under the product name and release code name specified. If sprint already exists with the given name, then returns its id.

By default the start date is today's date and end date is 1 week from the started date.

Returns sprint_id in xml format.

³⁵/₁₇ **Pre-Conditions**

Product must be present.

³⁵/₁₇ **Summary**

URL: <http://www.example.com/sprint/write>

Format: XML

Authentication: false

HTTP Method(s): POST

³⁵/₁₇ **Request**

HTTP POST Request Example

[session_id=1234&product_name=dummy_product&code_name=dummy_release&sprint_name=dummy_sprint](#)

³⁵/₁₇ **Response**

XML Example

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<sprint> <id>34</id></sprint>
```

OR

```
<davos><error> Sprint name not defined! </error></davos>
```

OR

```
<davos><error> Release code name not found! Release Code Name = dummy_release</error></davos>
```

OR

```
<davos><error> Release code name not defined! </error></davos>
```

OR

```
<davos><error> Product not found! Product Name = dummy_product </error></davos>
```

OR

```
<davos><error> Product name not defined! </error></davos>
```


DAVOS USER MANUAL		Pages (17/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

2.3.5 build/write

Creates/updates build with testplan_id and build_tag under latest sprint inside latest release.

Returns build_id in xml format.

³⁵₁₇ **Pre-Conditions**

Product, release, sprint must be present for the given test plan.

³⁵₁₇ **Summary**

URL: <http://www.example.com/build/write>

Format: XML

Authentication: false

HTTP Method(s): POST

³⁵₁₇ **Request**

HTTP POST Request Example

[session_id=1234&testplan_id=999&build_tag=myBuild](#)

³⁵₁₇ **Response**

XML Example

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<build><id>34</id></build>
```

OR

```
<davos><error> Build tag not defined! </error></davos>
```

OR

```
<davos><error> No Project phase exists for this release! Release Name =dummy_release</error></davos>
```

OR

```
<davos><error> No Release exists for this Product! Product Name = dummy_product</error></davos>
```

OR

```
<davos><error> Test Plan not found! Test Plan Id = 999 </error></davos>
```

OR

```
<davos><error>Test Plan Id missing!</error></davos>
```

DAVOS USER MANUAL		Pages (18/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

2.3.6 testplan/write

Creates testplan entry with the testplan_name under the product name, code name and sprint name specified. If test plan already exists with the given name, then returns its id.

³⁵/₁₇ **Pre-Conditions**

Product must be present.

³⁵/₁₇ **Summary**

URL: <http://www.example.com/testplan/write>

Format: XML

Authentication: false

HTTP Method(s): POST

³⁵/₁₇ **Request**

HTTP POST Request Example

session_id=1234&product_name=dummy_product&testplan_name=dummy_testplan

³⁵/₁₇ **Response**

XML Example

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<testplan> <id>34</id></testplan>
```

OR

```
<davos><error> Product not found! Product Name = dummy_product</error></davos>
```

OR

```
<davos><error> Product name not defined! </error></davos>
```

OR

```
<davos><error> Product does not exist! Product Id = dummy_product</error></davos>
```

DAVOS USER MANUAL		Pages (19/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

2.3.7 testsuite/write

Creates or updates a test suite entry with the product_name.

Returns testsuite_id in xml format.

³⁵/₁₇ **Pre-Conditions**

Product must be present.

³⁵/₁₇ **Summary**

URL:http://www.example.com/testsuite/write

Format: XML

Authentication: false

HTTP Method(s): POST

³⁵/₁₇ **Request**

HTTP POST Request Example

```
product_name=ABC&role_name=DavosClient&testsuite_name=WorkFlowSuite&comment=
description_of_the_testsuite&created_by=Username&modified_by=username&status=Planned&elapsedtime_min=100&elapsedtime_max=100&testsuitefilepath=path to
testsuite&initialization_time=100&execution_time=100&session_id=1234
```

³⁵/₁₇ **Response**

XML Example

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<testsuite><id>34</id></testsuite>
```

OR

```
<davos><error> Role Name does not exist! Role Name = </error></davos>
```

OR

```
<davos><error> Product not found! Product Name = </error></davos>
```

OR

```
<davos><error> Could not create testsuite </error></davos>
```

OR

```
<davos><error> Testsuite name not defined! </error></davos>
```

DAVOS USER MANUAL		Pages (20/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

2.3.8 testcase/write

Creates or updates a test case entry with the testsuite_id.

Returns testcase_id in xml format.

³⁵₁₇ **Pre-Conditions**

Test suite must be present.

³⁵₁₇ **Pre-Conditions**

Test suite must be present.

³⁵₁₇ **Summary**

URL:http://www.example.com/testcase/write

Format: XML

Authentication: false

HTTP Method(s): POST

³⁵₁₇ **Request**

HTTP POST Request Example

testsuite_id=78&testcase_identifier=TCS123&description=brief desc of the test case
&test_steps=test steps& verify_steps=verify steps&session_id=1234

³⁵₁₇ **Response**

XML Example

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<testcase><id>1234</id></testcase>
```

OR

```
<davos><error> Testsuite id not defined </error></davos>
```

OR

```
<davos><error> Testsuite does not exist! Testsuite Id = 78 </error></davos>
```

OR

```
<davos><error> testcase_identifier not defined! </error></davos>
```

OR

```
<davos><error> Could not update the existing testcase! </error></davos>
```

DAVOS USER MANUAL		Pages (21/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

2.3.9 testcycle/write

Creates or updates a testcycle with the testplan_id and build_id or tastpan_path.

If testcycle_id present then update the testcycle or if the session holding the testcycle id then it updates that testcycle. Else it will create a new testcycle.

Returns testcycle_id in xml format.

³⁵/₁₇ *Summary*

URL:http://www.example.com/testcycle/write

Format: XML

Authentication: false

HTTP Method(s): POST

³⁵/₁₇ *Request*

HTTP POST Request Example

initialization_time=100&execution_time=200&session_id=1234 (If only testcycle exists in that session)

OR

testplan_id=32&build_id=55&testcycle_id=89&initialization_time=100&execution_time=200&session_id=1234

OR

testplan_path=Product:Release:Sprint:Build:Testplan&initialization_time=100&execution_time=200&session_id=1234

OR

testplan_path=Product:Release:Sprint:Testplan&initialization_time=100&execution_time=200&session_id=1234

³⁵/₁₇ *Response*

XML Example

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<testcycle><id>1234</id></testcycle>
```

DAVOS USER MANUAL		Pages (22/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

OR

<davos><error> Cannot report multiple testcycle in a single session </error></davos>

OR

<davos><error> Test cycle not found! Test Cycle Id = 999 </error></davos>

OR

<davos><error> Build not associated with this test cycle. Please associate a build with this test cycle and run the test. Test Cycle Id =999 </error></davos>

OR

<davos><error> Build does not exist! Build Id = 178 </error></davos>

OR

<davos><error> No Release exists for this Product! Product Name = dummy_pro </error></davos>

OR

<davos><error> No matching Arguments found</error></davos>

DAVOS USER MANUAL		Pages (23/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

2.3.10 variables/write

Creates variable for a testcase with the testcase identifier, variable name and variable value

Returns variable_id in xml format

³⁵₁₇ **Pre-Conditions**

Product must be present.

³⁵₁₇ **Summary**

URL:http://www.example.com/variables/write

Format: XML

Authentication: false

HTTP Method(s): POST

³⁵₁₇ **Request**

HTTP POST Request Example

testcase_identifier=TC001&variable_name=xxx&variable_value=555&session_id=1234

³⁵₁₇ **Response**

XML Example

<?xml version="1.0" encoding="utf-8" ?>

<testcasevariablevalue> <id>1234</id></testcasevariablevalue>

OR

<davos><error> Testcase identifier not defined! </error></davos>

OR

<davos><error> Variable name not defined! </error></davos>

OR

<davos><error> Variable value not defined! </error></davos>

OR

<davos><error> Product does not exist! Product Name =dummy_pro </error></davos>

DAVOS USER MANUAL		Pages (24/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

OR

*<davos><error> Session does not hold any product details or testplan details!
</error></davos>*

OR

<davos><error> Testsuite does not exist in chosen Product! Testsuite Name = ts, Product Id = 999 </error></davos>

OR

<davos><error> Testsuite name not defined! </error></davos>

OR

<davos><error> Test case does not exist in the testsuite! </error></davos>

OR

<davos><error> Could not create testsuite variables </error></davos>

2.4 Posting execution Data

2.4.1 testexecutiondetails/write

Creates or Updates a test execution details entry with the testcycle_id and topologysset_id

Returns testexecutiondetail_id in xml format.

³⁵₁₇ **Summary**

URL: <http://www.example.com/testexecutiondetails/write>

Format: XML

Authentication: false

HTTP Method(s): POST

³⁵₁₇ **Request**

HTTP POST Request Example

```
testcycle_id=43&topologysset_id=9&role_name=zermatt-
server&testcase_identifier=TCS&test_engineer=Automation&status=Pass&comments=Nothi
ng&testexecution_time=100&performanceexecutiondetail_id=1&testsuite_name=WORKSSts
&session_id=1234
```

³⁵₁₇ **Response**

XML Example

DAVOS USER MANUAL		Pages (25/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

<?xml version="1.0" encoding="utf-8" ?>

<testexecutiondetails> <id>1234</id></testexecutiondetails>

OR

<davos><error> Testsuite name not defined! </error></davos>

OR

<davos><error> Testcase identifier not defined! </error></davos>

OR

<davos><error> Role does not match with the Testsuite Role! </error></davos>

OR

<davos><error> Cannot report multiple testcycle in a single session </error></davos>

OR

<davos><error> Testcycle id or name missing! </error></davos>

OR

<davos><error> Test Cycle does not exist! </error></davos>

OR

<davos><error> Topologysset id or name missing! </error></davos>

OR

<davos><error> Topologysset does not exist! </error></davos>

OR

<davos><error> Role name missing! </error></davos>

OR

<davos><error> Role Name does not exist! </error></davos>

DAVOS USER MANUAL		Pages (26/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

2.4.2 performancemetric/initialize

Stores performance metric data temporarily in data base with testexecutiondetail_id

³⁵₁₇ **Summary**

URL: http://www.example.com/performancemetric/initialize/{testexecutiondetails_id}

Format: XML

Authentication: false

HTTP Method(s): POST

³⁵₁₇ **URL Parameters**

testexecutiondetails_id = testexecutiondetails_id of the test case

³⁵₁₇ **Request**

HTTP POST Request Example:

cpu_stat=24&io_stat=4567&session_id=1234

2.4.3 performance/write

Creates or updates performance execution detail for a test execution detail record. Requires testexecutiondetails_id, key_name, key_value. If key_name exists for the execution record then it updates its value, else creates a fresh record.

chart_type, chart_title can be send as an argument to display chart in ZERMATT.

³⁵₁₇ **Summary**

URL: <http://www.example.com/performance/write>

Formats: Xml

Authentication: false

HTTP Method(s): POST

³⁵₁₇ **Request**

³⁵₁₇ HTTP POST Request Example

³⁵₁₇ [session_id=1234&testexecutiondetail_id=999&key_name=CPUUsage&value=98&chart_type=bar&chart_title=CPU Usage](http://www.example.com/performance/write?session_id=1234&testexecutiondetail_id=999&key_name=CPUUsage&value=98&chart_type=bar&chart_title=CPU Usage)

³⁵₁₇ **Response**

DAVOS USER MANUAL		Pages (27/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

XML Example:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<performance> <id>34</id></performance>
```

OR

```
<davos><error> Key_name name not defined! </error></davos>
```

OR

```
<davos><error> Testexecutiondetails identifier do not exist! </error></davos>
```

OR

```
<davos><error> Testexecution details identifier not defined! </error></davos>
```

OR

```
<davos><error> Could not create performance_key! </error></davos>
```

OR

```
<davos><error> Could not update existing performance execution detail!  
</error></davos>
```

OR

```
<davos><error> Value not defined! </error></davos>
```

DAVOS USER MANUAL		Pages (28/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

2.4.4 performancemetric/finalize

Calculate the quotient with the previous stored metric and store the results in the database
testexecutiondetail_id

³⁵₁₇ **Summary**

URL:http://www.example.com/performancemetric/finalize/{samplesize}

Formats: Xml

Authentication: false

HTTP Method(s): POST

³⁵₁₇ **URL Parameters**

samplesize(by default it is 1)

It is the value with which the difference will be divided

³⁵₁₇ **Request**

HTTP Request Example:

cpu_stat=26&io_stat=5678&session_id=1234

2.4.5 /setTestCycleTopologySetValues

Creates or updates machine specific attributes participating in any test environment. Accepts testexecutiondetail_id, machine_attribute and value. machine_name can also be used to report for a particular topology, if that machine belongs to the testing environment.

Returns topologyattribute_id in xml format.

³⁵₁₇ **Summary**

URL:http://www.example.com/setTestCycleTopologySetValues

Format: XML

Authentication: false

HTTP Method(s): POST

³⁵₁₇ **Request**

HTTP POST Request Example

session_id=1234&testexecutiondetail_id=999&name=RAM&value=1GB

DAVOS USER MANUAL		Pages (29/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

OR

session_id=1234&testexecutiondetail_id=999&machine_name=LADAKH&name=RAM&value=1GB

³⁵₁₇ **Response**

XML Example

<?xml version="1.0" encoding="utf-8" ?>

<setTestCycleTopologySetValues>542</setTestCycleTopologySetValues>

OR

<davos><error> Testexecutiondetail does not exist! </error></davos>

OR

<davos><error> Testexecutiondetail id not defined! </error></davos>

OR

<davos><error> Machine attribute name not defined! </error></davos>

OR

<davos><error> Machine does not belong to this topologyset. </error></davos>

OR

<davos><error> Error while updating machine attribute. </error></davos>

OR

<davos><error> Error while creating new machine attribute. </error></davos>

DAVOS USER MANUAL		Pages (30/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

2.5 Other helper calls

2.5.1 /updateMyMachine

Updates the machine ip address for a particular machine name in machine catalog table. Uses the request machine ip address as seen by davos and updates the machine. Mostly used by NYON to keep ZERMATT updated with its present state.

Returns machine_name in xml format

³⁵₁₇ **Pre-Conditions**

Machine must be present inside machinecatalog table with its actual name.

³⁵₁₇ **Summary**

URL:http://www.example.com/updateMymachine

Format: XML

Authentication: false

HTTP Method(s): POST

³⁵₁₇ **Request**

HTTP POST Request Example

session_id=1234&machine_name=LADAKH

³⁵₁₇ **Response**

XML Example

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<machine>LADAKH</machine>
```

OR

```
<davos><error> Machine name missing! </error></davos>
```

OR

```
<davos><error> Cannot update machine ip address! </error></davos>
```

OR

```
<davos><error> Machine name not found! </error></davos>
```

2.5.2 /renameFile

DAVOS USER MANUAL		Pages (31/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

Used in Zermatt while uploading test suite files. Accepts testsuite_id and file type as argument. Supported file_type are txt,zip,xls.

File type = New File Name

txt = product_id.testsuite_id.Description.org_filename

zip = product_id.testsuite_id.Archive.org_filename

xls = product_id.testsuite_id.Testsuite.org_filename

Returns modified name of the file in xml format

³⁵₁₇ **Pre-Conditions**

ZermattPathToPubDirectory must be defined in frameworkmetadata table

³⁵₁₇ **Summary**

URL:http://www.example.com/renameFile

Format: XML

Authentication: false

HTTP Method(s): POST

³⁵₁₇ **Request**

HTTP POST Request Example

session_id=1234&testsuite_id=999&file_type=xls

³⁵₁₇ **Response**

XML Example

<?xml version="1.0" encoding="utf-8" ?>

<file>888.999.Testsuite.myTs.xls</file>

OR

<davos><error> Testsuite id not defined </error></davos>

OR

<davos><error> Testsuite file type not defined! </error></davos>

OR

<davos><error> Testsuite does not exist! </error></davos>

OR

<davos><error> File not found! </error></davos>

OR

<davos><error> File type not supported! </error></davos>

DAVOS USER MANUAL		Pages (32/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

2.5.3 /sendmail

Accepts testcycle_id, job_id and mail_body as its argument.

Fetches the recipients list from testplan table.

Return success in xml format if successfully able to send mail to all the recipients.

³⁵₁₇ **Pre-Conditions**

SMTP-ServerName, SMTP-ServerPort, Account-Username, Account-Password must be defined in Frameworkmetadata table.

³⁵₁₇ **Summary**

URL:http://www.example.com/sendmail

Format: XML

Authentication: false

HTTP Method(s): POST

³⁵₁₇ **Request**

HTTP POST Request Example

session_id=1234&testcycle_id=999&job_id=1234567&mail_body=content

³⁵₁₇ **Response**

XML Example

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<sendmail>success</sendmail>
```

OR

```
<davos><error> Testcycle id or name missing! </error></davos>
```

OR

```
<davos><error> Test Cycle do not exist. </error></davos>
```

OR

```
<davos><error> Test plan do not hold any email address list. </error></davos>
```

OR

```
<davos><error> Email address cannot be blank! </error></davos>
```

OR

```
<davos><error> Job Id is empty! </error></davos>
```

OR

```
<davos><error> Mail body is empty! </error></davos>
```

OR

```
<davos><error> Test plan not found. </error></davos>
```


DAVOS USER MANUAL		Pages (33/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

2.5.4 /commitNyonJob

Very specific to Nyon. Accepts job_id, zermatt_session_id, status, testcycle_id, submission_time, user_name as its argument. Used to keep ZERMATT updated of the current state of NYON and jobs running in it.

Returns job_id in xml format

³⁵/₁₇ **Summary**

URL: <http://www.example.com/commitNyonJob>

Format: XML

Authentication: false

HTTP Method(s): POST

³⁵/₁₇ **Request**

HTTP POST Request Example

session_id=1234&job_id=456789&zermatt_session_id=yeh67hue5373h378d&status,=running&testcycle_id=999&submission_time=2013-06-28-14:20:40&user_name=MIKE

³⁵/₁₇ **Response**

XML Example

```
<?xml version="1.0" encoding="utf-8" ?>
<commitNyonJobs>456789</commitNyonJobs>
```

OR

```
<davos><error> message </error></davos>
```

DAVOS USER MANUAL		Pages (34/34)
Author : chayan hazra	Version: Davos Version 6.4.1	Date :27-07-2013

3 Test execution reporting without ZUG

To report a test result use the following steps:

1. Create a new session in davos using rest call **2.10 session/create** , and use the returned session_id in the following steps.
2. Create a new testcycle using rest call **2.12 testcycle/write** , Use the testcycle id returned instep 3
3. Create a new execution record using rest call **2.15 testexecutiondetails/write**
Provide all the required arguments.
Use **2.8 role/read** and **2.19 topologysset/read** to read/validate the role or the topologysset.
4. Call **2.13 testcycle/write** to report back the total execution time to run the complete test cycle.

NOTE : If davos returns this message 'session timedout' any where in the following rest call use 2.12 update/session