Script documentation: run\_zephyr\_automation\_cycle

[Introduction 2](#_Toc103891012)

[Prerequisites 2](#_Toc103891013)

[Running the script 2](#_Toc103891014)

# Introduction

This document describes the run\_zephyr\_automation\_cycle.py script. Its purpose is to deploy a set of Velocity executions specific to an existing Zephyr cycle and update its results accordingly in Jira where one or more issues will be opened in case of execution failures.

# Prerequisites

1. Testcases in Velocity have to be mapped to the Zephyr test instances by assigning the tags with the same ID as the ones in Zephyr;

* for example if a Zephyr cycle has tests with IDs VDBR-1, VDBR -2 and VDBR -3, then Velocity must have a testcase for each of these three IDs (the mapping has to be one-to-one, one tag should be assigned to at most one testcase in Velocity);
* to check if a testcase with a certain tag exists in Velocity, go to the Library -> Automation Assets page, and use the Tags filter:

Graphical user interface, text, email

Description automatically generated

* adding/editing tags for testcases can be made only through iTest

# Running the script

For the script to run successfully, it will need to have an available agent with Python execution capabilities. Verify this by going to Reports -> Velocity Agents and checking that there is an Online agent (green status) with the proper capability:

A screenshot of a computer

Description automatically generated

The following steps need to be followed in order to manually trigger a script execution:

1. Go to the Library -> Automation Assets page;
2. Search for run\_zephyr\_automation\_cycle in the Search box;
3. Click on the Run Automation Asset button;

Graphical user interface, text, email

Description automatically generated

1. Click on Run;
2. Click on the Parameters tab;

Graphical user interface, application

Description automatically generated

1. The page will list a set of parameters which will be used as arguments throughout the script execution. The parameters are listed and described in the following table:

|  |  |  |
| --- | --- | --- |
| Name | Description | Example |
| jira\_project\_key | * the Jira project key in which the Zephyr tests and cycle can be found | VDBR |
| jira\_project\_release\_name | * the Jira key from which the jira project version name can be obtained from | VDBR-139 |
| zephyr\_test\_cycle\_name | * the name of the Zephyr test cycle which will be executed * if it is set to “N/A”, then runlist\_name parameter must be completed | Cycle1 |
| story\_key\_for\_comment | * name of the Jira key in which the comment with the final result of the execution will be opened | VDBR-139 |
| zephyr\_build | * Zephyr build version | a.2 |
| runlist\_name | * name of the Velocity runlist which will be executed in case the zephyr\_test\_cycle\_name is not mentioned (set to “N/A”) | FTTx\_ZTE\_List |
| topology\_name | * name of the topology which will be used for the runlist execution | FTTH\_ZTE\_Base\_Topology\_2 |

1. The execution of the script will extract the IDs of the tests in the Zephyr cycle given by zephyr\_test\_cycle\_name, and start a Velocity runlist execution with the testcase which are matching the tag search criteria:

Graphical user interface, text, application, email

Description automatically generated

1. After each testcase execution is completed, the script will update the status of the Zephyr test with their respective result;
2. In case of testcase failure, a bug will automatically be opened in Jira with the name of the testcase and failure reason, and linked to the main story in Jira;
3. At the end of the runlist execution, a comment with the final results will be created in the Jira story, together with an attached HTML report which contains detailed information on the finished executions.