Integrating with VersionOne

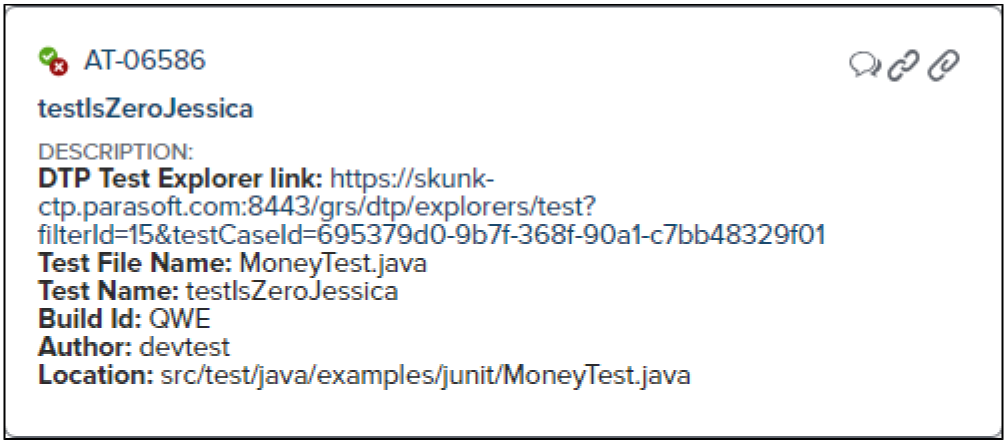
In this section:

Introduction

VersionOne is a popular browser-based platform for managing requirements and tracking issues. Parasoft DTP integrates with VersionOne, providing the following functionality:

- Ability to automatically update test results in VersionOne (see syncTestCases)
- Ability to automatically create defects and issues in VersionOne when DTP detects test failures (see syncTestFailures)
- Ability to manually create defects and issues in VersionOne from the Violations Explorer view
- Ability to manually create defects and issues in VersionOne from the Test Explorer view.

VersionOne assets include build and authorship information, as well as links back to the test or violation in DTP.



AT-06586

testIsZeroJessica

DESCRIPTION:
DTP Test Explorer link: https://skunk-ctp.parasoft.com:8443/grs/dtp/explorers/test?filterId=15&testCaseId=695379d0-9b7f-368f-90a1-c7bb48329f01
Test File Name: MoneyTest.java
Test Name: testIsZeroJessica
Build Id: QWE
Author: devtest
Location: src/test/java/examples/junit/MoneyTest.java

DTP includes an API for processing the test associations and sending the data to VersionOne. This enables you to automate this process so that, for example, you could add a job to your CI server that calls the API based on specific triggers in your environment.

You can associate multiple projects in DTP with a project in VersionOne, but you cannot associate the same DTP project with more than one VersionOne project.

This feature is currently in beta.

Configuration

The following sections describe how to create the relationships between Parasoft and VersionOne artifacts.

Connecting DTP to VersionOne Server

1. Choose **Report Center Settings** from the settings (gear icon) drop-down menu.
2. Choose **External Application [Beta]** from the Administration sidebar and choose **VersionOne** from the Application Type drop-down menu.
3. Enable the **Enabled** option.
4. Enter a name for your instance of VersionOne in the Name field.
5. Enter the VersionOne URL in the Application URL field.
6. The Display URL field is rendered in DTP interfaces when links to your VersionOne system are created.
7. Enter login credentials in the Username and Password field. The login must have sufficient privileges to create VersionOne issues in the projects specified in the Project Associations section.
8. Click **Test Connection** to verify your settings and click **Save**.

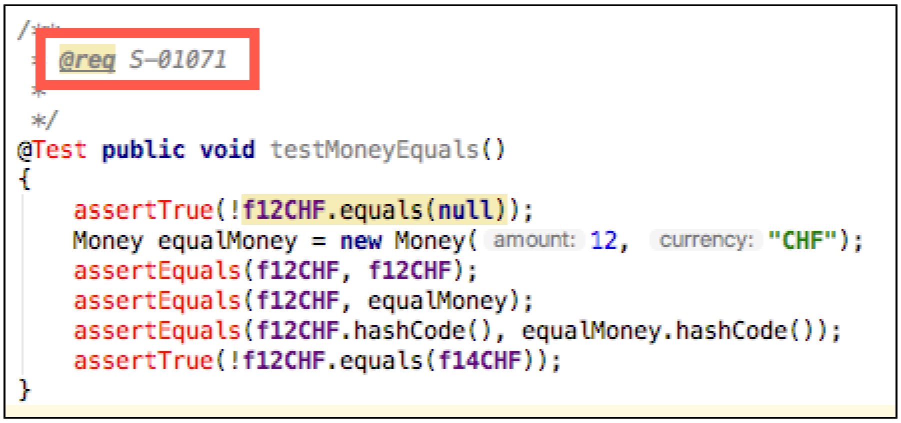
Associating DTP Projects with VersionOne Projects

1. Click **Create Project Association** and choose a project from the DTP Project drop-down menu in the overlay.
2. Enter the name of a VersionOne project in the External Project field and click **Create** to save the association.

Click the trash icon to remove a project association. Removing the project association does not remove links. If a new association is created, existing links between violations and VersionOne issues will be reactivated.

Associating Parasoft Test Cases with VersionOne Work Items

VersionOne work items are assets that represent a story, defect, or test set. Add the `@test` annotation and VersionOne test ID to the test code executed by your Parasoft tool to associate Parasoft tests with VersionOne tests. Add the `@req` annotation and VersionOne story ID in the test code to associate Parasoft tests with VersionOne stories. Refer to your Parasoft tool documentation for details on adding associations and the VersionOne documentation for information about getting work item IDs.



```
/**
 * @req S-01071
 *
 */
@Test public void testMoneyEquals()
{
    assertTrue(!f12CHF.equals(null));
    Money equalMoney = new Money( amount: 12,  currency: "CHF");
    assertEquals(f12CHF, f12CHF);
    assertEquals(f12CHF, equalMoney);
    assertEquals(f12CHF.hashCode(), equalMoney.hashCode());
    assertTrue(!f12CHF.equals(f14CHF));
}
```

You can make several associations, but VersionOne tests and tests executed by Parasoft have a one-to-one relationship. If multiple Parasoft tests are associated with a single VersionOne test, DTP will send results for the multiple tests to same VersionOne test.

Creating Defects and Issues

You can create defects and issues in VersionOne using the Test Explorer interface or call the API to them programmatically create them when tests fail. You can also create defects and issues associated with static analysis violations from the Violations Explorer. Additionally, if the VersionOne story is closed, DTP will still create the artifact, regardless of whether the API or UI is used.

VersionOne terminology

"Issues" in VersionOne refer to assets that highlight items that require project-wide visibility. Issues can be represent anything that is impeding or might impede the project team's delivery. "Defects" are assets that identify a discrepancy between the expected and actual behavior of a completed feature. Defects can be scheduled into sprints/iterations the same way stories/backlog items are or, they can be tracked as a separate work queue outside of any sprint.

The following chart describes how assets are created in VersionOne:

	Associated with @req annotation	Associated with @test annotation
Defect	• DTP creates assets in the Breaks Work Item field. • Links to the Test Explorer view	• DTP creates asset in the Breaks Work Item field. • Links to the Test Explorer view • Links to test in VersionOne
Issue	• DTP creates asset in the Blocking Issues field • Links to the Test Explorer	• DTP creates asset in the Blocking Issues field • Links to the Test Explorer view • Links to test in VersionOne

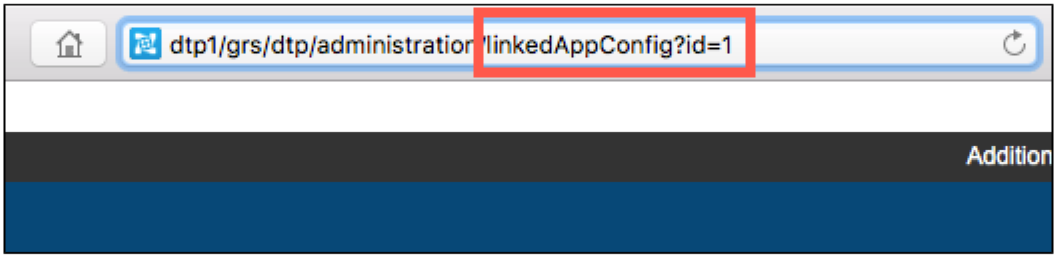
Using Explorer Views

The Test Explorer and Violations Explorer views enable you to create issues and defects for any test and violation, respectively, regardless of status. Refer to the following sections for details on creating VersionOne assets in explorer views:

- Violations Explorer > Creating an Issue in a Third-party System
- Test Explorer > Creating an Issue in a Third-party Systems

Using the API

You will need the configuration ID associated with your VersionOne integration to use the API. After configuring your connection to VersionOne, the ID will appear in the URL off the External Applications screen.



dtp1/grs/dtp/administration/linkedAppConfig?id=1

The following API endpoints are available.

syncTestCases

This endpoint adds or updates existing VersionOne tests that are associated Parasoft tests.

1. Execute your tests and send the reports to DTP.
2. After DTP receives the results, send a POST method call the API:

```
curl -u <username>:<password> -X "POST" "https://<host>:<port>/grs/api/v1.7/linkedApps/configurations/
{configurationId}/syncTestCases?filterId=<xyz>&buildId=<abc>"
```

DTP will process the report and send the data to VersionOne. The response will include the VersionOne tests that were created or updated with the test results data. The test results are ignored if DTP is unable to find the test requirement in VersionOne, which may occur if the incorrect `@test` or `@req` annotation is was added to the test code or if the item was deleted in VersionOne.

```
{
    "created" : [
    ],
    "updated" : [
        "Test:1545 for AT-01053, testName = test_quoteGhsLine_Exp_Act_3",
        "Test:1546 for AT-01054, testName = test_quoteGhsLine",
        "Test:1554 for AT-01056, testName = test_quoteGhsLine_Exp_Act_10",
        "Test:7177 for S-01045, testName = test_quoteGhsLine_moreThanOne"
    ],
    "ignored" : [
    ]
}
```

syncTestFailures

This endpoint create defects in VersionOne when DTP collects test failures associated with VersionOne tests. If a test fails for multiple builds, multiple defects will be created.

```
curl -u <username>:<password> -X "POST" "https://<host>:<port>/grs/api/v1.7/linkedApps/configurations/
{configurationId}/syncTestFailures?filterId=<xyz>&buildId=<abc>"
```

DTP will process the report and send the data to VersionOne. The response will include the failed tests and associated VersionOne tests.

```
[
    {
        "testCase" : {
            "id" : "a43a74f2-0d29-372f-8509-12bb8f41b5d5",
            "test" : {
                "name" : "testAppMainDemo1"
            }
        },
        "linkId" : "D-01273"
    },
    {
        "testCase" : {
            "id" : "40ab50ff-af9f-3f5b-b773-ecf35385dda4",
            "test" : {
                "name" : "testAppMainDemo2"
            }
        },
        "linkId" : "D-01274"
    },
    {
        "testCase" : {
            "id" : "29636230-1c65-3349-b6ad-ac0f999b18aa",
            "test" : {
                "name" : "testAppMainforDemo3"
            }
        },
        "linkId" : "D-01275"
    }
]
```