

NestedFlow Automation – Integration With Test Management Tools

Creation Date: 9/15/2023

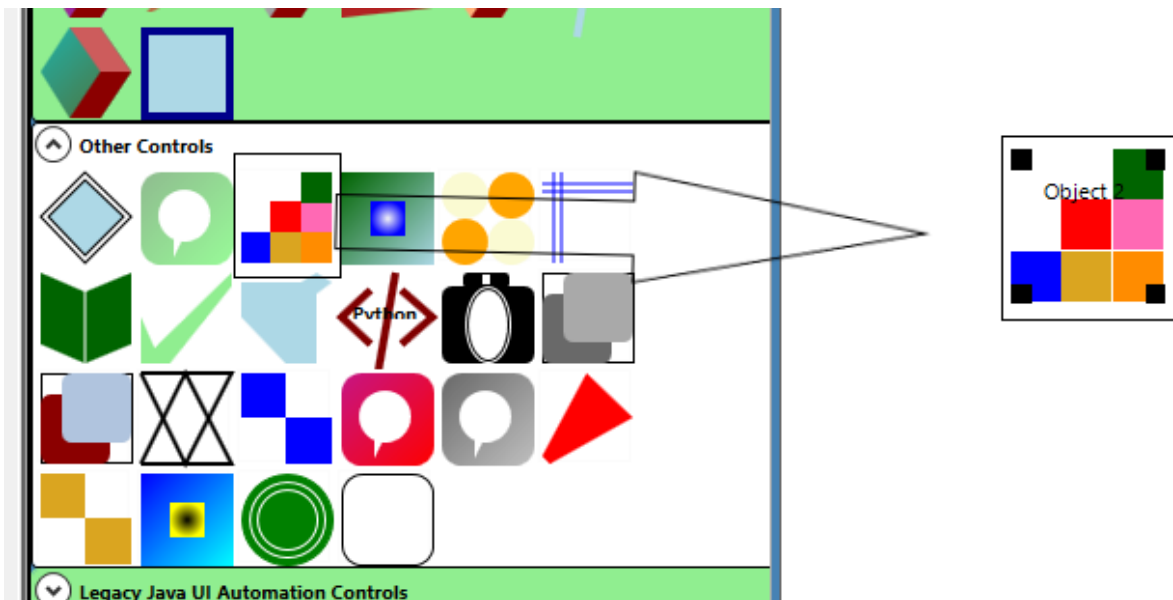
Table Of Contents

Introduction	1
ALM OTA Access	2
Post Update Script.....	4
Script:	5

Introduction

This document explains the ways in which the Test results can be integrated with Test Management tools (e.g.: ALM, Zephyr Scale etc.). This is a key requirement for any test automation tool which will help in reporting the findings of automated testing effectively. Tool has inbuilt ALM OTA integration feature but also can be integrated with JIRA or other Test management tools which provide API based access.

ALM OTA Access



ALMUPDATE component is available in Other Controls section.

Right Click on the component and choose option → **ALM Test Execution Update Option**

ALM integration form opens:

A screenshot of the 'Test Execution Integration' dialog box. It features a blue background and several input fields. The fields are: URL (a long text box), ALM User (with 'palle' entered), Password, Domain, Project, Test Lab Folder, Test Set Name, and Test Instance. Below these fields are radio buttons for 'Status: Pass', 'Status: Fail', 'Attach Log', and 'Derive'. At the bottom, there is a 'Log Files' section with a text box and 'Ok' and 'Cancel' buttons.

URL → ALM URL (usually ending with qcbn)

ALM User → user name

Password → ALM login password

Domain → ALM Domain Name

Project → ALM Project Name

Test Lab Folder → Test Lab folder id (not the path but folder id)

Test Set Name → Test Set Name

Test Instance → Test Instance Name

If the status intended is **Pass** then click: **Status: Pass**

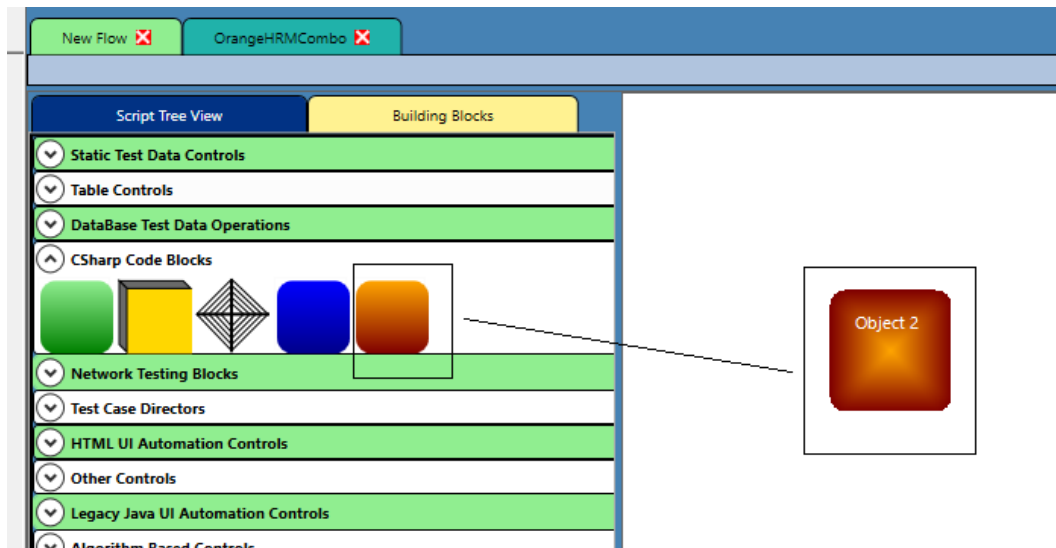
If the status intended is **Fail** then click: **Status: Fail**

Derive (This option is depreciated. It doesn't do anything)

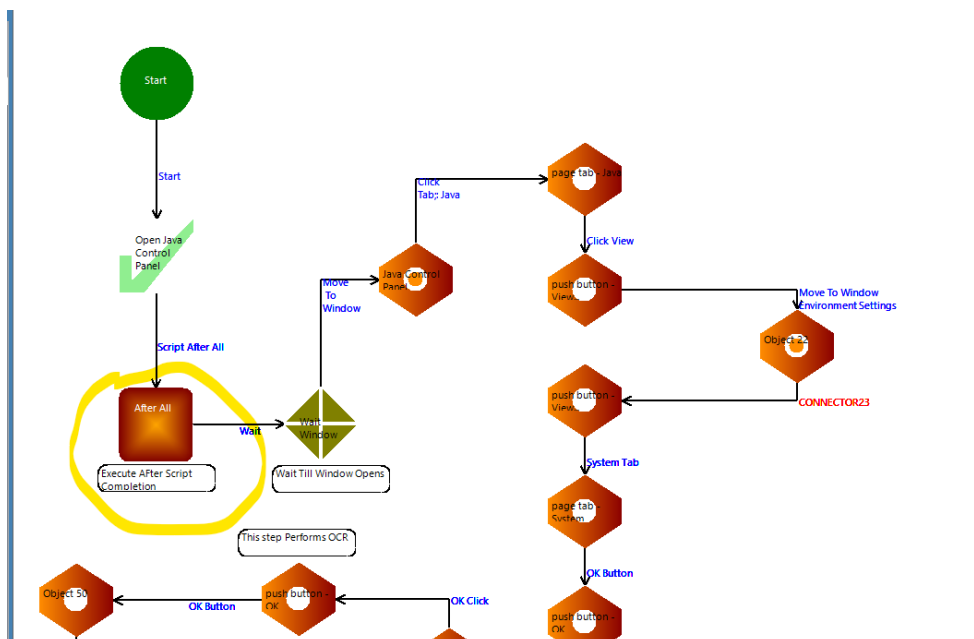
Attach Log: If chosen it will attach the execution log to test instance. Also, it will attach all files in Log Files section separated by “;”

Post Update Script

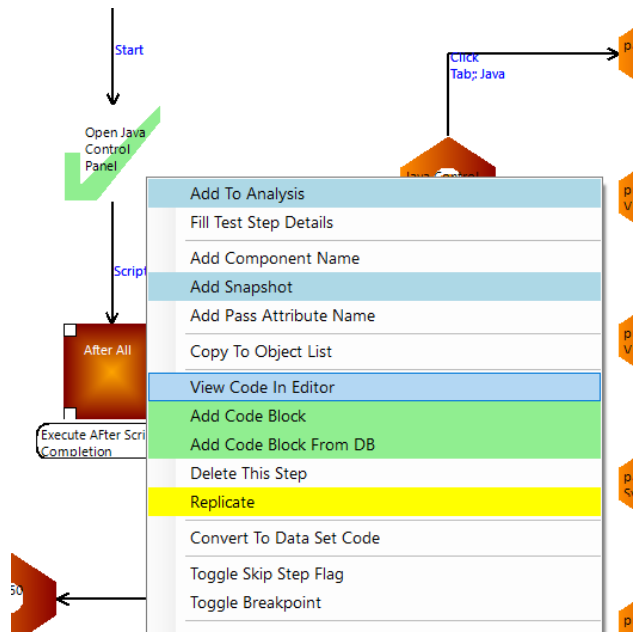
NestedFlow Automation provides Post script event. It can be used to perform Test Management updates.



For example, given script below: In which we are defining a post script event. It will be executed only after entire script is executed



Add Code Block or **Add Code Block From DB** option can be used to assign a script



Below is a sample script to update test status in Zephyr Scale

Script:

```
using System;
using System.Xml;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;
using System.Drawing;
using System.Net;
using System.IO;

//Return value from Eval Code will be assigned to column
//DataTable T1: Whatever Data in Bottom Grid Of Query 1
//string s1 : Credential> User Name
//string s2 : Credential> Password
//string s3 : Credential> User Name
//string s4 : Credential> Password

namespace NSUpdateTable
{
    public class CSUpdateTable
    {
        //Return value from Eval Code will be assigned to column
        public static void Main()
        {
        }
        public static DataTable UpdateTable(DataTable T1,string s1,string s2,string s3,string s4,ref string s5)
        {
            try
            {
                string _URL = "https://api.zephyrscale.smartbear.com/v2/testexecutions";
                string _TOKEN = "<TOKEN>";
                string _projectKey="PRJ";
```

```

string _TestCaseKey="PRJ-XXX";
string _TestCycleKey="PRJ-YYYY";
string _Status="Fail";
string REQBODY = "{" + "\r\n";
    REQBODY = REQBODY + "\"projectKey\": \"" + _projectKey + "\", " + "\r\n";
    REQBODY = REQBODY + "\"testCaseKey\": \"" + _TestCaseKey + "\", " + "\r\n";
    REQBODY = REQBODY + "\"testCycleKey\": \"" + _TestCycleKey + "\", " + "\r\n";
    REQBODY = REQBODY + "\"statusName\": \"" + _Status + "\", " + "\r\n";
    REQBODY = REQBODY + "\"testScriptResults\": [ " + "\r\n";
    REQBODY = REQBODY + "\"statusName\": \"Pass\", " + "\r\n";
    REQBODY = REQBODY + "\"actualEndDate\": \"" + DateTime.Now.ToString("yyyy-MM-ddTHH:mm:ssZ") + "\", " + "\r\n";
    REQBODY = REQBODY + "\"actualResult\": \"Website Opened successfully\" " + "\r\n";
    REQBODY = REQBODY + "}, " + "\r\n";

    REQBODY = REQBODY + "\"statusName\": \"Pass\", " + "\r\n";
    REQBODY = REQBODY + "\"actualEndDate\": \"" + DateTime.Now.ToString("yyyy-MM-ddTHH:mm:ssZ") + "\", " + "\r\n";
    REQBODY = REQBODY + "\"actualResult\": \"Account Number field visible successfully\" " + "\r\n";
    REQBODY = REQBODY + "}, " + "\r\n";

    REQBODY = REQBODY + "\"statusName\": \"Pass\", " + "\r\n";
    REQBODY = REQBODY + "\"actualEndDate\": \"" + DateTime.Now.ToString("yyyy-MM-ddTHH:mm:ssZ") + "\", " + "\r\n";
    REQBODY = REQBODY + "\"actualResult\": \"Account Number field Entered successfully\" " + "\r\n";
    REQBODY = REQBODY + "}, " + "\r\n";

    REQBODY = REQBODY + "\"statusName\": \"Pass\", " + "\r\n";
    REQBODY = REQBODY + "\"actualEndDate\": \"" + DateTime.Now.ToString("yyyy-MM-ddTHH:mm:ssZ") + "\", " + "\r\n";
    REQBODY = REQBODY + "\"actualResult\": \" Web Page Opened successfully\" " + "\r\n";
    REQBODY = REQBODY + "}, " + "\r\n";

    REQBODY = REQBODY + "\"statusName\": \"Pass\", " + "\r\n";
    REQBODY = REQBODY + "\"actualEndDate\": \"" + DateTime.Now.ToString("yyyy-MM-ddTHH:mm:ssZ") + "\", " + "\r\n";
    REQBODY = REQBODY + "\"actualResult\": \"Resources Opened successfully\" " + "\r\n";
    REQBODY = REQBODY + "}, " + "\r\n";

    REQBODY = REQBODY + "\"statusName\": \"Pass\", " + "\r\n";
    REQBODY = REQBODY + "\"actualEndDate\": \"" + DateTime.Now.ToString("yyyy-MM-ddTHH:mm:ssZ") + "\", " + "\r\n";
    REQBODY = REQBODY + "\"actualResult\": \" resource 1 Shown successfully\" " + "\r\n";
    REQBODY = REQBODY + "}, " + "\r\n";

    REQBODY = REQBODY + "\"statusName\": \"Pass\", " + "\r\n";
    REQBODY = REQBODY + "\"actualEndDate\": \"" + DateTime.Now.ToString("yyyy-MM-ddTHH:mm:ssZ") + "\", " + "\r\n";
    REQBODY = REQBODY + "\"actualResult\": \"Resource 2 displayed\" " + "\r\n";
    REQBODY = REQBODY + "}, " + "\r\n";

    REQBODY = REQBODY + "\"statusName\": \"Pass\", " + "\r\n";
    REQBODY = REQBODY + "\"actualEndDate\": \"" + DateTime.Now.ToString("yyyy-MM-ddTHH:mm:ssZ") + "\", " + "\r\n";
    REQBODY = REQBODY + "\"actualResult\": \"Resource 3 displayed\" " + "\r\n";
    REQBODY = REQBODY + "}, " + "\r\n";

    REQBODY = REQBODY + "\"statusName\": \"Pass\", " + "\r\n";
    REQBODY = REQBODY + "\"actualEndDate\": \"" + DateTime.Now.ToString("yyyy-MM-ddTHH:mm:ssZ") + "\", " + "\r\n";
    REQBODY = REQBODY + "\"actualResult\": \"Logged out successfully\" " + "\r\n";
    REQBODY = REQBODY + "}" + "\r\n";

    REQBODY = REQBODY + "]" + "\r\n";
    REQBODY = REQBODY + "}" + "\r\n";

    HttpWebRequest myHttpWebrequest = (HttpWebRequest)HttpWebRequest.Create(_URL);
    myHttpWebrequest.Method = "POST";
    myHttpWebrequest.KeepAlive = true;

```



```

myHttpRequest.Headers.Add("Authorization", "Bearer " + _TOKEN);
byte[] data = System.Text.Encoding.ASCII.GetBytes(REQBODY);
myHttpRequest.ContentType = "application/json";
myHttpRequest.Accept = "application/json";
myHttpRequest.ContentLength = data.Length;
Stream requestStream = myHttpRequest.GetRequestStream();
requestStream.Write(data, 0, data.Length);
requestStream.Close();

HttpWebResponse myHttpResponse = (HttpWebResponse)myHttpRequest.GetResponse();
Stream responseStream = myHttpResponse.GetResponseStream();
StreamReader myStreamReader = new StreamReader(responseStream, System.Text.Encoding.Default);
string responseContent = myStreamReader.ReadToEnd();
myStreamReader.Close();
responseStream.Close();
myHttpResponse.Close();
}
catch
{
}
return (T1);
}
}
}

```

