# INTEGRATING TESTDRIVE


TECHNICAL GUIDE

# TABLE OF CONTENTS

## INTRODUCTION

This document describes how to integrate TestDrive automated testing with other lifecycle tools, such as a CI/CD platform.

The primary executable component of TestDrive is the Playlist. A Playlist controls the execution of an automated test suite and provides information about the outcome of the test execution.

## EXECUTING A TESTDRIVE PLAYLIST

A Playlist can be executed from a command line such as the Windows Command Prompt or Windows PowerShell.

The command to execute a Playlist is in two parts. The first is the path to the TestDrive executable and the second is a list of arguments that tells TestDrive which Playlist to execute and how to access it.

For example:

```
C:\Program Files (x86)\Original Software\Build 7063\OriginalSoftware.TestDrive.TestDrive.exe
 /connection:b0b90ff1-038d-496c-942d-a68610320c86
 /playlist:0c744972-62f3-4554-a197-af5c2b8d5879
 /applicationdefinitionid:b4f9aa06-6754-4353-9378-a6e5f5d647d0
 /applicationinstanceid:cbbfeb09-b280-449f-8617-bae019b598e8
 /save:True
```

The /connection: parameter points to repository database connection details held locally on an execution PC. Please note that this may vary between PCs. (See Issue Mitigation for further details)

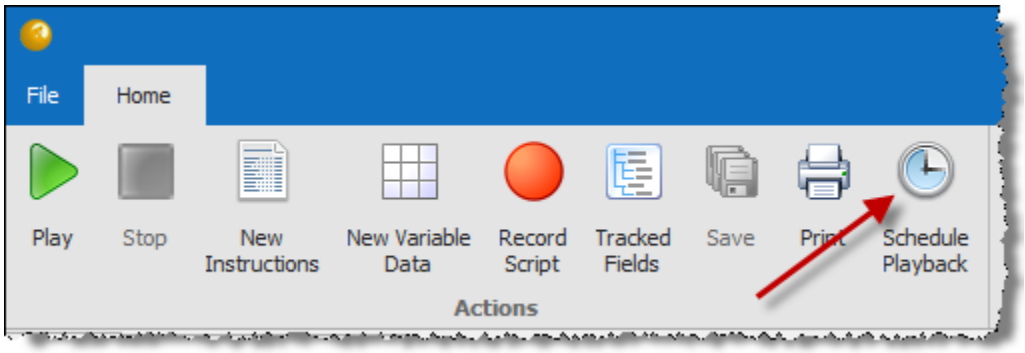
The /playlist: parameter is the unique ID of the Playlist. This value will always be the same for a specific Playlist.

The /applicationdefinitionid: and /applicationinstanceid: hold the unique ids of the organisational structures within the repository database where the Playlist is stored. These values will always be the same unless the logical location of the Playlist is changed.

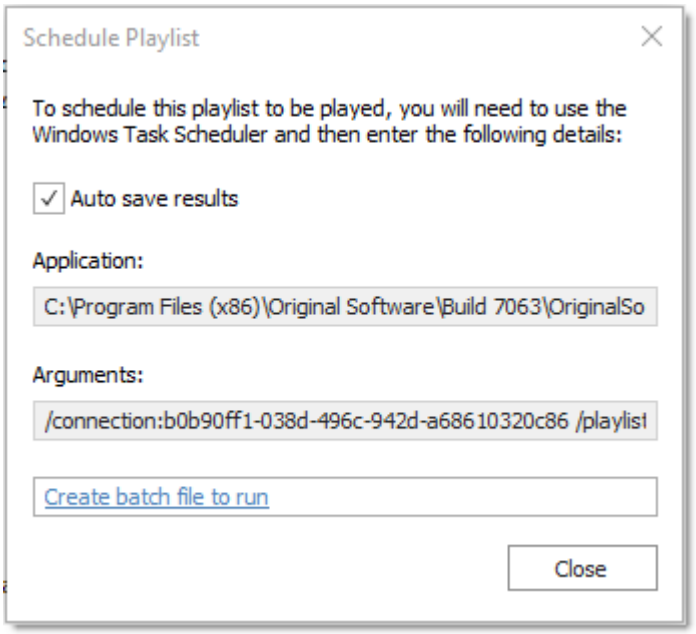
The /save: parameter defines whether or not the execution results should be saved.

## DETERMINING THE PLAYLIST EXECUTION ARGUMENTS

The TestDrive Playlist editor provides details of the TestDrive executable path, and the arguments required to execute the Playlist.




Select **Schedule Playback** from the editor toolbar. This will display the **Schedule Playlist** dialog.




The **Application** section contains the TestDrive executable path details and the **Arguments** section contains all of the necessary values for finding the Playlist. The **Auto save results** checkbox state will determine whether the /save: parameter is set to True or False.

You can also create a Windows batch file from the dialog.

## DETERMINING THE OUTCOME OF A TEST EXECUTION

The TestDrive executable will pass back a 'return code' indicating whether or not it was able to find and launch the specified Playlist. This code will have a value of **99** if the launch fails. Aside from that, the executable does not provide any feedback as to the result of the test execution. Instead, TestDrive provides data, in the form of XML files, that contain that information. What data is provided is configured within the Playlist. The data that is available is as follows:

- Run Header
- Start/Finish Time
- Run Details
- Run Details with Counts
- Pass/Fail
- Performance
- HTML Size
- Images Size
- URLs
- Checking Rule Summary
- Check Count

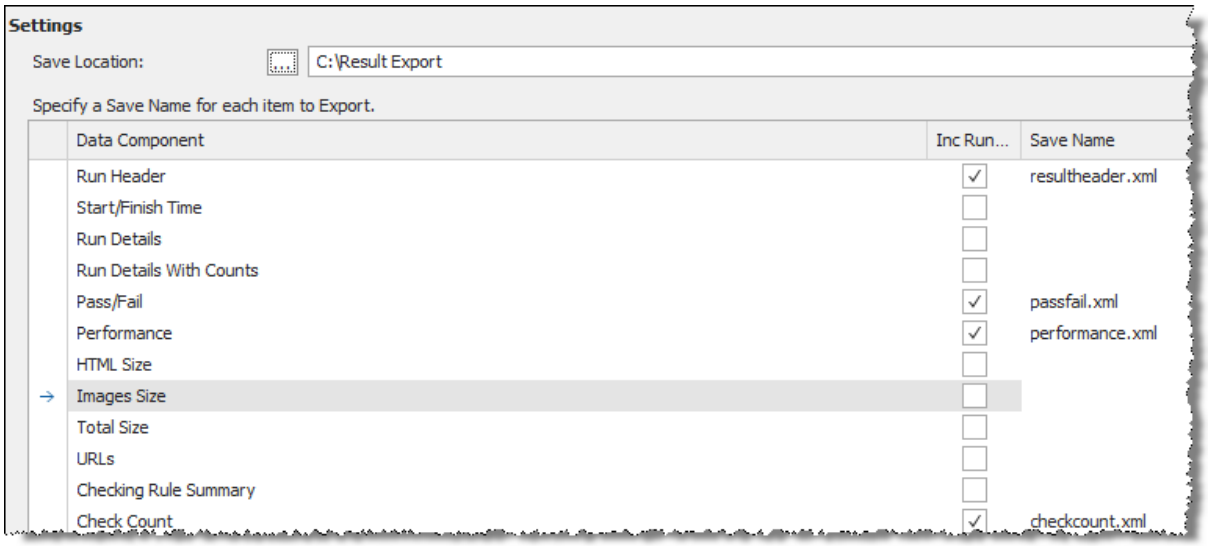
There are some more advanced data sets available if TestDrive is run in its Advanced Web Statistics mode.

## CONFIGURING WHAT DATA TO PROVIDE

The Playlist Export Data post-playback action defines what data is provided and where it is saved. It also provides the ability to condition if that data is provided.

The Export Data post-playback action requires a location to save the data, in the form of XML files. This can be any location accessible to the PC such as mapped network drive, a local drive, or a mapped SharePoint location.

Whether a data is provided for a particular category is set simply by defining a file name. In addition, the file name can have the unique TestDrive Run ID included in the name.




## SAMPLE RESULT DATA

### Run Header

```
<?xml version="1.0" encoding="utf-8"?>
<AnalyticsRunHeaderData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <PassFailStatusAsText>Fail</PassFailStatusAsText>
  <Authenticated>false</Authenticated>
  <RunId>Run 206</RunId>
  <What>Bakery Demo - Chromium - Result Export</What>
  <Screens>Played 22 screens</Screens>
  <Elapsed>1 minute 55 seconds</Elapsed>
  <InfoCount>0</InfoCount>
  <WarnCount>55</WarnCount>
  <ErrorCount>1</ErrorCount>
  <CriticalCount>0</CriticalCount>
</AnalyticsRunHeaderData>
```

## Pass/Fail


```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfAnalyticsPassFailData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <AnalyticsPassFailData>
    <StatusAsText>Fail</StatusAsText>
    <AuthenticatedAsBoolean>false</AuthenticatedAsBoolean>
    <DateTime>2021-07-09T10:18:27</DateTime>
    <User>Bower, Tim</User>
    <Comment>Execution Started - Set default Run Status
Incorrect Card Discounts
Finished playback successfully</Comment>
  </AnalyticsPassFailData>
</ArrayOfAnalyticsPassFailData>
```


## Check Counts


```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfAnalyticsCheckCountData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <AnalyticsCheckCountData>
    <ScriptId>4c2a8023-c46f-429a-b503-54fae168d85f</ScriptId>
    <EventId>4023ffa7-c50d-4263-840d-5e1c10a2916a</EventId>
    <Description>1: Original Bakery - Login</Description>
    <Information>0</Information>
    <Warning>0</Warning>
    <Error>0</Error>
    <Critical>0</Critical>
  </AnalyticsCheckCountData>
  <AnalyticsCheckCountData>
    <ScriptId>27657e9a-d864-4341-9d64-96122a481bdb</ScriptId>
    <EventId>c1587484-2022-4e8c-9a86-0d4e826e1aad</EventId>
    <Description>1: Original Bakery - Home</Description>
    <Information>0</Information>
    <Warning>7</Warning>
    <Error>0</Error>
    <Critical>0</Critical>
  </AnalyticsCheckCountData>
</ArrayOfAnalyticsCheckCountData>
```

# ISSUE MITIGATION

## INCORRECT CONNECTION ID

Connection strings are held locally on each PC in the OSMConnections.xml file that is stored in the **%appdata%\Original Software\Original Software Manager** folder. Each connection has a unique identifier.

If a connection has been configured individually on each PC, this ID will differ between PCs. To prevent this, create the connection on one PC and then copy the OSMConnections.xml file to each of the other PCs so that they are identical.


Original Software