opentext™

UFT One

Software version: All versions

Runtime Engine Guide

Go to Help Center online

https://admhelp.microfocus.com/uft/



Document release date: October 2024

Send Us Feedback



Let us know how we can improve your experience with the Runtime Engine Guide.

Send your email to: admdocteam@opentext.com

Legal Notices

© Copyright 2024 Open Text.

The only warranties for products and services of Open Text and its affiliates and licensors (“Open Text”) are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Disclaimer
Certain versions of software accessible here may contain branding from Hewlett-Packard Company (now HP Inc.) and Hewlett Packard Enterprise Company. This software was acquired on September 1, 2017 by Micro Focus and is now offered by OpenText, a separately owned and operated company. Any reference to the HP and Hewlett Packard Enterprise/HPE marks is historical in nature, and the HP and Hewlett Packard Enterprise/HPE marks are the property of their respective owners.

About the UFT One Runtime Engine

Overview

OpenText™ UFT One Runtime Engine enables you to run UFT One tests (both GUI and API) and business process tests on your computer without installing the entire UFT One IDE. In addition, you can install the Runtime Engine without the Run Results Viewer, UFT One Add-in for ALM, sample applications, or Help documentation. This can potentially save you valuable disk space on your computer.

When you run the test with the Runtime Engine, you can access and run the test from a number of different places without the need to open the UFT One interface and configure UFT One options. When the test runs it runs in the background. At the end of the test run, you can view the test results.

Using the Runtime Engine requires very little experience in using UFT One. You do not need to edit tests, change settings, or understand how to make UFT One work with your application. You simply select the test, run the test, and view the run results.

Applicable scenarios

Runtime Engine can be used in a number of different scenarios:

Running tests and components from ALM	You can set up test runs from the Test Lab module in ALM, and run these on a computer using the Runtime Engine. This enables you to run the test, without the need to interact with the UFT One interface (such as loading UFT One add-ins in the Add-in Manager dialog box).
Running tests from automation	You can run tests using automation using the Runtime Engine. The Runtime Engine installation enables you to save disk space on the computer running these tests, freeing up system resources on that computer for other tasks.

Running tests using the Jenkins plug-in	Runtime Engine can be installed on a build server or computer running builds of your applications. Using the Jenkins plug-in, you run a UFT One test as a post-build action of your application's build process. Having Runtime Engine installed on this computer to run UFT One tests enables you to free up system resources for the important application build tasks.
Running tests using external UFT One tools	When you install the Runtime Engine, you have external tools which enable you to run UFT One tests locally, including the Test Batch Runner and the Silent Test Runner. These tools enable you to run a test locally against your application as it is developed, and view the results instantly after the test run. As Runtime Engine does not enable you to edit a test, this version of the UFT One installation can be used by your application's developers and QA on an ongoing basis to provide regular testing of the application throughout the development process.

Runtime Engine Settings options

Runtime Engine also supports all the UFT One Add-ins as the full UFT One IDE, so you can run tests using any supported technology using the Runtime Engine. All objects and methods for all UFT One Add-ins are supported for use with the Runtime Engine.

As part of running a test, you can set specific run-time options. These options are set in the Runtime Engine Settings dialog box, available from the Windows Start menu:

Add-ins	You can specify add-ins to be loaded.
Run result export options	You can specify how and what the Runtime Engine should export from the run results after a run session.
Autosave options	You can specify whether to automatically save the copies of open GUI tests every few minutes and the auto-save interval.

Remote connection options	• You can specify if other applications are permitted to run tests on this computer using the Runtime Engine.
	• You can specify whether to run tests continuously on a remote computer even after the remote session is disconnected.
	  • Specify the resolution to use on the remote computer after an RDP session is disconnected from the remote computer.
	  • Use the **Poll to Detect Disconnect** option and set the polling frequency when your tests fail to run on a remote disconnected machine.
	  This option enables UFT One to regularly detect a disconnected session instead of waiting for a disconnect event notification. The polling frequency ranges from 10 to 60 seconds.
	Note: To modify these options, you must have a UFT One license installed.

Run options	You can specify how the Runtime Engine runs tests, including the format of the run results, whether the run results are opened automatically after a test run, and if the Runtime Engine takes screen captures or movies of the run session.

Text Recognition options	You can specify how the Runtime Engine works with text in your application when running a GUI test. When setting the **Text Recognition > ABBYY OCR Language** option, you must use values from the following list: **Available values** Abkhaz, Adyghe, Afrikaans, Agul, Albanian, Altaic, ArmenianEastern, ArmenianGrabar, ArmenianWestern, Awar, Aymara, AzeriCyrillic, AzeriLatin, Bashkir, Basque, Belarusian, Bemba, Blackfoot, Breton, Bugotu, Bulgarian, Buryat, Catalan, Chamorro, Chechen, ChinesePRC (for ChineseSimplified), ChineseTaiwan (for ChineseTraditional), Chukcha, Chuvash, Corsican, CrimeanTatar, Croatian, Crow, Czech, Danish, Dargwa, Digits, Dungan, Dutch, DutchBelgian, English, EskimoCyrillic, EskimoLatin, Esperanto, Estonian, Even, Evenki, Faeroese, Fijian, Finnish, French, Frisian, Friulian, GaelicScottish, Gagauz, Galician, Ganda, German, GermanLuxembourg, GermanNewSpelling, Greek, Guarani, Hani, Hausa, Hawaiian, Hebrew, Hungarian, Icelandic, Ido, Indonesian, Ingush, Interlingua, Irish, Italian, Japanese, Kabardian, Kalmyk, KarachayBalkar, Karakalpak, Kasub, Kawa, Kazakh, Khakas, Khanty, Kikuyu, Kirgiz, Kongo, Korean, Koryak, Kpelle, Kumyk, Kurdish, Lak, Lappish, Latin, Latvian, Lezgin, Lithuanian, Luba, Macedonian, Malagasy, Malay, Malinke, Maltese, Mansi, Maori, Mari, Maya, Miao, Minankabaw, Mohawk, Mongol, Mordvin, Nahuatl, Nenets, Nivkh, Nogay, Norwegian, NorwegianBokmal, NorwegianNynorsk, Nyanja, Occidental, Ojibway, OldEnglish, OldFrench, OldGerman, OldItalian, OldSpanish, Ossetic, Papiamento, PidginEnglish, Polish, PortugueseBrazilian, PortugueseStandard, Provencal, Quechua, RhaetoRomanic, Romanian, RomanianMoldavia, Romany, Ruanda, Rundi, Russian, RussianOldSpelling, Samoan, Selkup, SerbianCyrillic, SerbianLatin, Shona, Sioux, Slovak, Slovenian, Somali, Sorbian, Sotho, Spanish, Sunda, Swahili, Swazi, Swedish, Tabassaran, Tagalog, Tahitian, Tajik, Tatar, Tinpo, Tongan, Tswana, Tun, Turkish, Turkmen, Tuvin, Udmurt, UighurCyrillic, UighurLatin, Ukrainian, UzbekCyrillic, UzbekLatin, Visayan, Welsh, Wolof, Xhosa, Yakut, Zapotec, Zulu The **ABBYY OCR Language** option is available only if you include the ABBYY OCR Engine feature in the UFT One installation.

Web and Windows Application options	You can specify how the Runtime Engine runs tests for specific scenarios against a Web application or Windows application.

See also:

- "How to Use the UFT Runtime Engine" on the next page
- "Prerequisite - How to set test run options" on page 11

How to Use the UFT Runtime Engine

The UFT One Runtime Engine is a slimmer installation of UFT One that enables you to run UFT One tests without the entire UFT One IDE, such as in the following scenarios:

- Running tests and components from ALM (either from the Test Plan, Test Lab, or Business Components modules)
- Running tests from an automation script
- Running tests from the Jenkins plugin
- Running tests from external UFT One tools such as the Test Batch Runner or the Silent Test Runner
- Running tests locally against your application (after it has been updated) without needing to edit the test

What do you want to know?

- "Prerequisite - How to ensure the Runtime Engine is installed correctly" below
- "Prerequisite - How to install your license" on the next page
- "Prerequisite - How to set test run options" on page 11
- "How to run tests from ALM" on page 12
- "How to run tests using automation" on page 13
- "How to run tests from the Jenkins plugin" on page 15
- "How to run rests using the Test Batch Runner" on page 16
- "How to run tests using the Silent Test Runner" on page 17

Prerequisite - How to ensure the Runtime Engine is installed correctly

Before running a test using the Runtime Engine, ensure the following:

- You have installed and loaded the correct add-ins for your test. The currently loaded add-ins can be modified in the Runtime Engine settings dialog box.

- You have selected the **Allow other products to run tests and components** option in the UFT One Configurations screen of the installation wizard

If you did not perform one or both of these options, you can configure this option by running the UFT One Setup again and selecting the **Change** option in the installation wizard.

Prerequisite - How to install your license

Before you can run tests with the Runtime Engine, you must install a valid UFT One license. A license is required also for modifying the Remote Connection options in the Runtime Engine settings.

You do this either using the license installation wizard or with the command line:

From the License Wizard	1. Open the **License Installation Wizard** (available from the Windows Start menu). 2. Follow the steps in the wizard to install the correct license type. **Note:** • When requesting a license, you need the license type **UFT One Runtime Engine**. • For a seat license, you need a license key string or a license key file. • For a concurrent license installation, you need the address of the License Server on which your licenses are installed. 3. Exit the wizard. When the Runtime Engine runs your test, it will check for the license before running the test.
From the Command Line	1. Open a command window. 2. In the command window, enter one of the following: • For a seat license: **"<UFT_One_installdir>\bin\HP.UFT.LicenseInstall.exe" seat "<license key string>"/seat "<path to the license key file>"**

> **Note:** You need to provide either the license key string or the path to the license key file, not both.

- For a concurrent license: **"<UFT_One_installdir>\bin\HP.UFT.LicenseInstall.exe" concurrent <license ID> <license version> <server name/address> [<redundant server name/address>] [/force]**

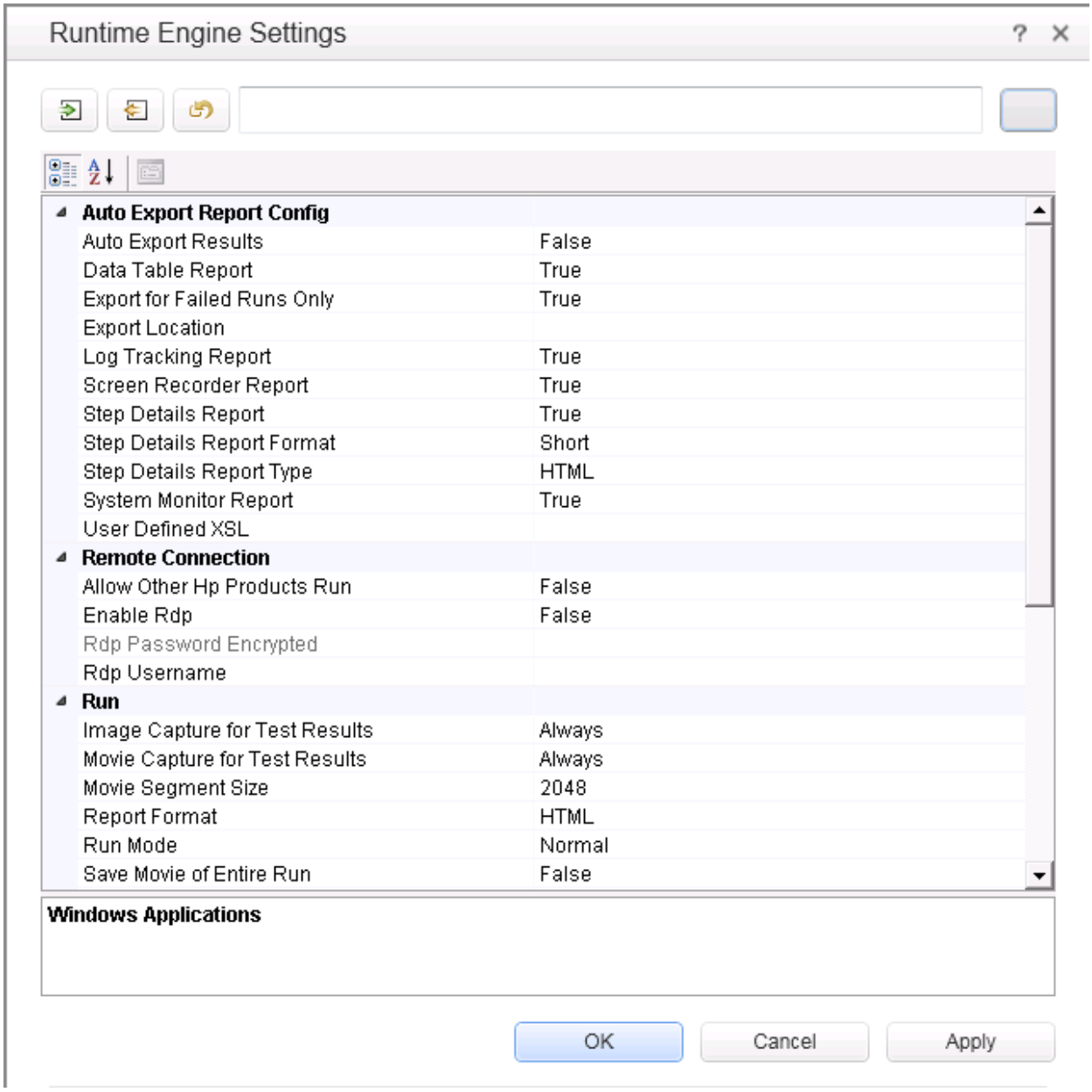
  > **Note:**
  >
  > - The **<server name/address>** or **<redundant server name/address>** uses the format **server name/address:port**. The port number is optional.
  > - The **<server name/address>** or **<redundant server name/address>** must be in the same format as the server name or address listed in the **Main** tab of the Configuration pane of the License Server. For more details on setting up the License Server address, see the AutoPass License Server online documentation.
  > - Both the **[redundant server name/address]** and **/force** parameters are optional.
  > - The **/force** parameter saves the license installation information even if the current installation fails. On subsequent UFT One sessions, UFT One checks the listed license server for the listed license.

3. Exit the wizard.

When the Runtime Engine runs your test, it will check for the license before running the test.

Prerequisite - How to set test run options

Before you run your tests, you should specify the options for running your tests. You can use the Runtime Engine Settings dialog box to specify these settings. This dialog box is available from the Windows Start menu.



You can update the settings in the following ways:

Manually configure the settings:	Using the options provided in the dialog box, select the relevant values from the drop-down lists and click **Apply** to save the settings. **Note:** Before saving, make sure that UFT One is not running. After you update these settings, you can click the **Export Settings to Script** button , which exports these settings into an automation script file (saved with a **.vbs** extension). This script can be reimported on your computer for later test runs or shared with others running tests using the Runtime Engine.
Import the settings from a configuration script:	You can also update these settings from a generated automation script exported from the UFT One IDE on another computer or exported from another installation of the Runtime Engine. To import exported settings: 1. Click the **Import Settings from Script** button . 2. Navigate to the directory where your import script (with a **.vbs** extension) is saved and click **Open**. The settings in the dialog are updated accordingly to reflect the data in the script.

For descriptions of each of the options, select the item in the dialog box and view the description at the bottom of the dialog.

> **Note:** If you are setting the options for the Remote Desktop connection in the Runtime Engine Settings dialog, you must log in to the computer as an administrator and start the HP.UFT.Helper service before changing these options.

How to run tests from ALM

Using the Runtime Engine, you can run tests and components directly from the Test Lab Modules:

1. Ensure that the DCOM has been configured to run tests from ALM as part of the installation.

If this configuration was not done during the Runtime Engine installation, you can perform the configuration with the Additional Installation Requirements tool (installed with the Runtime Engine), available from the Windows Start menu. Use the **Configure DCOM Settings for ALM Integration** option.

2. Set your test run options, as described in "Prerequisite - How to set test run options" on page 11.
3. From the Start menu , select **Remote Agent** .
4. In the Windows tray, right-click the **Remote Agent** icon  and select **Settings**.
5. In the Remote Agent Settings dialog box, configure the following options as needed:
   - **Restart testing tool after __ runs**
   - **Close testing tool after __ idle minutes**

   > **Note:** The other options in the Remote Agent are not applicable for use with the Runtime Engine.

6. In ALM, from the Test Plan module, run individual tests or a test set, depending on your needs. For details on setting up test sets and running from ALM, see the *Application Lifecycle Management User Guide*.
7. After the test run is finished, view the results as specified in your test run options.

How to run tests using automation

Automation is a Microsoft technology that makes it possible to access software objects inside one application from other applications. These objects can be created and manipulated using a scripting or programming language such as VBScript or VC++. Automation enables you to control the functionality of an application programmatically.

Using automation, you can write scripts that open UFT One, set options and settings, and run the test. This enables you to automatically run tests at scheduled times without your manual intervention.

For example, you can create and run an automation script from Microsoft Visual Basic that does the following:

- Loads the required add-ins for a test or component
- Starts UFT One
- Opens the test or component
- Configures settings that correspond to those in the Options dialog box, Test or Business Components Settings dialog box, and the Record and Run Settings dialog box (all in the full UFT One IDE)
- Runs the test or component

When you run tests using automation, do the following:

1. **If another computer is running automation scripts via the computer with the Runtime Engine**, ensure that the DCOM has been configured to run tests with automation as part of the installation.

   If this configuration was not done during the Runtime Engine installation, you can perform the configuration with the Additional Installation Requirements tool (installed with the Runtime Engine), available from the Windows Start menu. Use the **Configure DCOM Settings for Automation Scripts** option.
2. Ensure that your script is accessible from the computer with the Runtime Engine.
3. Set your test run options, as described in "Prerequisite - How to set test run options" on page 11.
4. Run your test script, using your preferred automation run model. For details on running tests using automation, see the UFT One Help Center.

> **Note:** The options in your automation scripts that set UFT One to be visible are ignored when running a test with the Runtime Engine. The test runs without problems even if your automation scripts contain these options, including the following automation methods:
>
> - AutoGenerateWith
> - DisableVORRecognition

> - DisplayKeywordView
> - TimetoActivateWinAfterPoint
> - WithGenerationLevel

5. After the test run is finished, view the results as specified in your test run options.

How to run tests from the Jenkins plugin

The Jenkins CI plugin integrates with the Runtime Engine to enable you to automatically run your UFT One tests directly from the file system as part of the build process.

To run tests from the file system as part of the build:

1. Ensure that your tests or test batches are saved and accessible on the file system.
2. Set your test run options, as described in "Prerequisite - How to set test run options" on page 11.
3. Configure the Jenkins plug-in to run functional tests. For details, see UFT One and Jenkins.
4. In the Jenkins plug-in, specify the location of the tests or test batches in the **Tests** box of the plug-in.

   > **Note:** To specify multiple entries, click the down arrow on the right of the field and enter each test path on a separate line. Make sure that the paths are accessible from all machines in the local network.

   When a build of your application runs, the Jenkins plug-in will automatically run this test as configured in the plug-in settings.
5. After the test run is finished, view the results as specified in your test run options.

How to run rests using the Test Batch Runner

You can also use the Test Batch Runner (installed with the Runtime Engine) to run a single test or a group of tests combined into a test batch (saved with an **.mtb** extension).

To run tests using the Test Batch Runner:

1. Ensure that your tests and test batches are saved and accessible on the file system.
2. Set your test run options, as described in "Prerequisite - How to set test run options" on page 11.
3. Open the **Test Batch Runner** (available from the Windows Start menu).
4. In the Test Batch Runner main window, add tests to the list of tests to be run in one of the following ways:
   - Select **File > Add** and select a precompiled Test Batch (**.mtb**) file.
   - Create or open a test batch file (**File > New/Open**), and then add tests to the test batch (**Tests > Add Test from Folder**).
   - **UFT One 24.2 and later:** Drag tests from your file system into the Test Batch Runner. You can drag specific tests, or a test folder. If you drag a test folder, all the tests in the folder are added to the batch.
5. Select the check boxes of the tests you want to run.

   The tests run according to the order in which you selected them.

   UFT One 24.2 and later:
   - The **Order** column displays the order in which you selected the tests, which is the order in which the tests run.
   - (Optional) Click the **Random** button to instruct UFT One to run the selected tests in a random order.
   - Drag and drop individual tests in the Test Batch Runner window to change the order in which the tests are displayed.
6. When you have added all the necessary tests to run, in the toolbar, click the **Run** button .

The Runtime Engine runs the test or tests on your application automatically. You can view the status of each individual test run in the main Test Batch Runner window.

7. After the test run is finished, view the results as specified in your test run options. The Test Batch Window provides the location of the run results.

How to run tests using the Silent Test Runner

The Runtime Engine enables you to run tests using the Silent Test Runner (installed with the Runtime Engine). This enables you to simulate the way a UFT One test runs from LoadRunner Professional and Application Performance Management and to verify that your UFT One test is compatible with LoadRunner Professional and Application Performance Management.

> **Note:** If you want to run tests with the Silent Test Runner, you must install the **Samples** component in the **Custom Setup** screen of the installation wizard.

To run a test with the Silent Test Runner:

1. Ensure that your tests and test batches are saved and accessible on the file system.
2. Set your test run options, as described in "Prerequisite - How to set test run options" on page 11.
3. Open the **Silent Test Runner** (available from the Windows Start menu).
4. In the Silent Test Runner window, click the **Browse** button and select a test.
5. Click **Run Test** to start the test run.

   When the test run is complete, the Silent Test Runner window displays a message confirming the end of the test run.
6. In the Silent Test Runner window, click the **Test Run Log** window. A text file is displayed with a report of all test steps from the current run.