OpenText Functional Testing

Software version: 26.1

Installation Guide

Go to Help Center online

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

Document release date: January, 2026

Send Us Feedback

Let us know how we can improve your experience with the Installation Guide.
Send your email to: admdocteam@opentext.com

Legal Notices

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.

Contents

Install 7
Overview 7
Installation guide content 7
Printer-friendly guide 8

Demos and compliance 9
Demo applications 9
Accessibility 9
Unicode Compliance 10

Enterprise deployment 11
User Account Control (UAC) 11
Stingray or Terminal Emulator Add-ins 12

Installation packages 14
Full OpenText Functional Testing installation package 14
Smaller OpenText Functional Testing installation packages 15
Additional OpenText Functional Testing-related installations 16

Before you install 20
Installation prerequisites 20
Required access permissions 23
Required permissions for OpenText Functional Testing 23
Required permissions for OpenText Application Quality Management 24
Required permissions for BPT 25

Installation wizard 26
Download the installation package 26
Run the installation wizard 27
Product improvement program 27
Change the user interface language 28
Custom Setup screen 28
OpenText Functional Testing Configuration screen 30

Silent installation 33
Before running a silent installation 33
Silent commands for installing prerequisites 34
Install OpenText Functional Testing silently 37
Include specific features 38

Set configuration options 43
Additional command properties 45
Install localized versions of OpenText Functional Testing 46
Install Web 2.0 Add-ins 46
Install a standalone OpenText Functional Testing Add-in for ALM 47

Install the Package for the Web Bundle 48
Overview 48
Download the Package for the Web Bundle 49
Install OpenText Functional Testing using an installation wizard 49
Run installation from a command line interface 50

Upgrade 53
Before you upgrade 53
Perform your upgrade 53
Configuration file location 54
Notes for upgrading 55

Verify the installation 59
Run the Installation Check Tool analysis 59
Learn an Installation Check Tool report 59

Install Web 2.0 add-ins or extensibility toolkits 61

Before connecting to OpenText Application Quality Management 62
For Microsoft Windows 10 & 11, Windows Server 2016 & 2019 62
For Microsoft Server 2012 63
Enable UAC again if required 63

Known issues when installing 65
Files in use 65
Component registration failed 66
Change or repair the OpenText Functional Testing installation 66
OpenText Functional Testing installations and other ADM software 66
OpenText Functional Testing installations and Microsoft software 68
Installing the OpenText Functional Testing Agent browser extension 71
OpenText Functional Testing installations and 64-bit applications 72
OpenText Functional Testing installations and Java 72
Installing in languages other than English 73

Licensing 75

License types 75
View license information 76
AutoPass License Server 76
Seat vs. concurrent licenses 77
Seat licenses 77
Concurrent licenses 78
License editions 79
Supported license editions 79
Licensing fallback mechanism 80
Manage licenses with the wizard 82
Manage licenses with the command line 89
Run the License Installer from the command line 89
Define seat licenses with the command line 89
Consume concurrent licenses with the command line 90
Configure license behavior 92
General license settings 93
Configure the licensing fallback mechanism 93
Configure licensing timeouts 95
Licensing FAQs 96
OpenText Functional Testing Help Center licensing scope 96
Which license should I install? 97
How do I install the AutoPass License Server? 97
If I am using concurrent licenses, how do I connect to the License Server? 97
How do I install licenses if I am deploying OpenText Functional Testing across an enterprise network? 98
How do I manage the concurrent licenses on the License Server? 98
Can I configure license behavior myself? 98
Can I use the AutoPass License Server with a proxy? 99
What is a cleanup license? 99
My demo license is expiring early. What can I do? 99
Known issues with licensing 100
Security Reference 103
Installing and using your product securely 103
Installation and deployment security 105
DCOM configuration settings 105
Browser policy settings 107
Connecting to OpenText Application Quality Management 108
Securing test information 109
GUI testing 109
API testing 110
Working with Safari on Mac 110
Connecting to OpenText Functional Testing Lab 111

Connecting to a Remote AI Object-Detection Service 111
Connecting to Cloud OCR vendors 112
CI server integration 113

Download Help Center 115
Access your local help by default 115

Install

This installation help provides information about installing and upgrading OpenText Functional Testing, as well as managing and using supported license types.

Overview

OpenText™ Functional Testing is a unified functional testing product solution for functional test and regression test automation, combined with API testing.

To check for software updates, patches, or service packs for OpenText Functional Testing, visit the Software Support Site.

> **Note:** The information in this guide is also available in the OpenText Functional Testing Online Help Center.

Installation guide content

The following table describes where to find different types of information in this Installation guide.

Install tasks	Details
Overview information	"Demos and compliance" on page 9 "Enterprise deployment" on page 11
Installation packages	"Installation packages" on page 14
Before you install	"Before you install" on page 20 "Installation prerequisites" on page 20 "Required access permissions" on page 23

Install tasks	Details
Installation steps	"Installation wizard" on page 26 "Silent installation" on page 33 "Install the Package for the Web Bundle" on page 48 "Verify the installation" on page 59 "Install Web 2.0 add-ins or extensibility toolkits" on page 61
Upgrading	"Upgrade" on page 53
Troubleshooting	"Known issues when installing" on page 65
OpenText Application Quality Management integration	"Before connecting to OpenText Application Quality Management" on page 62
Licensing	"Licensing" on page 75
Security Reference	"Security Reference" on page 103

Printer-friendly guide

The installation guide is also available in PDF: OpenText Functional Testing Installation Guide

Demos and compliance

This topic provides background information on using OpenText Functional Testing.

In this section:

- Demo applications 9
- Accessibility 9
- Unicode Compliance 10

Demo applications

Many examples in this guide use the Advantage Online Shopping sample web site. The URL for this web site is: https://www.advantageonlineshopping.com/#/.

> **Note:** You must register a user name and password to use this site. These credentials are not persistent. If you start a new session on the Advantage Online Shopping sample web site, you need to redefine your credentials.

A sample Flight Windows-based application is also provided with the OpenText Functional Testing installation.

You can open the **Flight API** and **Flight GUI** applications in various ways:

- From the Windows Start menu.
- **<Installdir>/samples/Flights Application/FlightsGUI.exe** (for the Flight GUI application)
- **<Installdir>/samples/Flights Application/FlightsAPI.exe** (for the Flight API application)

Accessibility

Many operations are performed using the mouse.

In accordance with Section 508 of the W3C accessibility standards, OpenText Functional Testing also recognizes operations performed using the **MouseKeys** option in the Windows Accessibility Options utility.

Additionally, you can perform many operations using shortcut keys.

Unicode Compliance

OpenText Functional Testing is Unicode compliant according to the requirements of the Unicode standard, enabling you to test applications in many international languages.

Test non-English language applications as long as the relevant Windows language support is installed on the OpenText Functional Testing computer.

Names and paths of tests and resources (for example, function libraries, object repositories, and recovery scenarios) are not Unicode compliant and therefore should be specified in English or in the language of the operating system.

Enterprise deployment

When installing OpenText Functional Testing in an enterprise business model across many computers in a network or company, ensure that you have administrator permissions on each computer.

OpenText Functional Testing also supports silent installations. For details, see "Silent installation" on page 33.

In this section:

- User Account Control (UAC) 11
- Stingray or Terminal Emulator Add-ins 12

User Account Control (UAC)

You do not need to turn off the computer's User Account Control (UAC).

For users to connect to OpenText Application Quality Management from OpenText Functional Testing for the first time without turning off UAC, you must also install the ALM client MSI file on each machine.

Generate a custom MSI for all your users with the ALM Client MSI Generator. This tool enables you to configure the settings for your ALM servers before installing the client-side MSI.

Configure the settings for a custom MSI, as described in the user guide provided with each MSI Generator version.

> **Note:** When configuring the settings, select the **Check Include Component Registration** and **Use Shared Deployment Mode** options.

Stingray or Terminal Emulator Add-ins

If your users are using either the Stingray or Terminal Emulator add-ins, there are additional configurations you or the user must perform after installing OpenText Functional Testing.

Both the Stingray and Terminal Emulator Add-ins

Run the Additional Installation Requirements, available in the Start menu, after the main installation on each computer.

In the Additional Installation Requirements, select the **Run Stingray Wizard** and/or **Run Terminal Emulator Wizard** options, and follow the configuration wizard steps to set up the add-ins.

Stingray Add-in

After installing OpenText Functional Testing, users must run the Stingray Support Configuration Wizard from inside OpenText Functional Testing: **Tools > Options > GUI Testing** tab > **Stingray** pane > **Select Version**.

This configuration requires no administrative permissions.

Terminal Emulator Add-in

After installing OpenText Functional Testing, users must run the Terminal Emulator configuration wizard from inside OpenText Functional Testing: **Tools > Options > GUI Testing** tab > **Terminal Emulator** pane **> Open Wizard**

Users must have administrator permissions to run this wizard.

You can also run the wizard once, and then save the settings to a registry file that can be deployed on all computers, as follows:

1. In the final screen of the Terminal Emulator Wizard, select the **Save terminal emulator settings to file** option.

> **Note:** Ensure that you record the vendor name and emulator name assigned to the configuration, as well as the exact location of the **.reg** file.

2. Copy the file to the **<Installdir>\dat** folder on your computer.
3. Double-click the registry file to open the Registry Editor message box.
4. Click **Yes** to add the information into the registry. A message opens confirming that the information has been copied into the registry.
5. Click **OK**. The emulator name assigned to this configuration is added to the list of available terminal emulators for your OpenText Functional Testing installation.

Installation packages

This section provides details about the OpenText Functional Testing installation packages available for OpenText Functional Testing.

You can access the installation packages from the free trial page or by selecting your account on the Software Licenses and Downloads page.

In this section:

- Full OpenText Functional Testing installation package 14
- Smaller OpenText Functional Testing installation packages 15
- Additional OpenText Functional Testing-related installations 16

Full OpenText Functional Testing installation package

The Full UFT One DVD Release includes the following content:

- The OpenText Functional Testing setup program.

  During the OpenText Functional Testing installation, you can specify which features and add-ins to include. This controls both the product functionality and the installation size.

  For a full list of included features, see "Custom Setup screen" on page 28.
- OpenText Functional Testing prerequisites
- "Additional OpenText Functional Testing-related installations" on page 16

Smaller OpenText Functional Testing installation packages

Smaller installation packages are available for OpenText Functional Testing:

Package	Content
Package for the Web Bundle	This compressed, lighter installation package provides the OpenText Functional Testing setup program only. This includes the same features and add-ins as the Full UFT One DVD Release. OpenText Functional Testing installation prerequisites are included but not automatically installed. This provides a faster installation if you already have all the prerequisites installed. **Excluded:** "Additional OpenText Functional Testing-related installations" on the next page
Core UFT One DVD Bundle	This package provides a slightly faster and smaller installation if you do not use ABBYY for text recognition and the remote AI Object-Detection service. **Excluded:** • ABBYY OCR Engine files. These files are required if you want to include the ABBYY OCR Engine feature in the OpenText Functional Testing installation. The files are included in the Full UFT One DVD Release and the Package for the Web Bundle. • Remote AI Object-Detection service package. This is a separate installation program included only in the Full UFT One DVD Release. **Notes:** • Recognizing AITableCell objects in AI table containers requires ABBYY text recognition. • To add the ABBYY OCR Engine to your installation later, download the **UFT OCR Expansion Pack**.

Package	Content
UFT OCR Expansion Pack	Provides ABBYY OCR Engine support for the **Core UFT One DVD Bundle**. **Contains:** Only the ABBYY OCR Engine files **Note:** Download the version that matches the Core UFT One DVD Bundle version you use.

Additional OpenText Functional Testing-related installations

The Full UFT One DVD Release and Core UFT One DVD Bundle packages provide the OpenText Functional Testing setup program, as well as additional setup programs for other standalone OpenText Functional Testing-related installations.

To run one of the additional installation programs, perform the following steps:

1. Unzip the package you downloaded.
2. Run the **Setup.exe** file.
3. Select the program you want to install from the OpenText Functional Testing installation wizard **Startup** screen.

Additional OpenText Functional Testing-related installations include:

Install	Description
OpenText Functional Testing Add-in for Application Quality Management	Enables OpenText Functional Testing to communicate with OpenText Application Quality Management and run tests or components. Install the standalone version only if OpenText Functional Testing is not installed on the computer. To install this together with OpenText Functional Testing, select to install the **ALM Plugin** while installing OpenText Functional Testing. If you did not install this originally with OpenText Functional Testing, and want to later on, run the installation wizard again. Select **Change**, and then select **ALM Plugin**in the Custom Setup screen.
Extensibility SDKs	Enable you to develop support for Java, .NET, WPF, or Web object not supported by OpenText Functional Testing by default.
Web 2.0 Toolkit Support	Enables you to recognize and use objects in your test from Web 2.0 technologies, including: • ASP .NET Ajax • Dojo • GWT (Google Web Toolkit) • jQueryUI • Salesforce Lightning • SiebelOpenUI • EXT-JS • YahooUI Web 2.0 Toolkits appear in OpenText Functional Testing as GUI Add-ins.
License server setup	Enables you to install the AutoPass License Server, which is used to install and manage OpenText Functional Testing concurrent and commuter licenses. For details, see "Licensing" on page 75 and the AutoPass License Server online documentation.

Install	Description
Run Results Viewer setup	Enables you to install a standalone version of the Run Results Viewer. Install the standalone version only if OpenText Functional Testing is not installed on the computer.
OpenText Functional Testing for Developers setup	Enables you to install OpenText Functional Testing for Developers, a functional testing tool that enables you to code tests directly from your developer IDE. • Install the standalone version only if OpenText Functional Testing is not installed on the computer. • To install this together with OpenText Functional Testing, select to install it while installing OpenText Functional Testing. If you did not install this originally with OpenText Functional Testing, and want to later on, run the installation wizard again. Select **Change**, and then select the **OpenText Functional Testing for Developers** in the Custom Setup screen. • Before installing OpenText Functional Testing for Developers, you must install a supported version of **Node.js**. For a list of **Node.js** versions supported for OpenText Functional Testing for Developers, see the Support Matrix. For more details, see the *OpenText Functional Testing for Developers Help Center*.
Remote AI Object-Detection Service setup **(Included only in the Full UFT One DVD Release)**	Enables you to install the remote AI Object-Detection service on one central powerful computer. Then this service can provide AI object-detection services to all your OpenText Functional Testing computers. Before you install, make sure you meet all the prerequisites for this service, as described in the OpenText Functional Testing Help Center.

See also:

- "Before you install" on page 20
- "Licensing" on page 75

- "Before connecting to OpenText Application Quality Management" on page 62

Before you install

Before you install OpenText Functional Testing, perform the steps below.

> **Caution:** OpenText Functional Testing can potentially be used to record user actions and/or network communications. Therefore, we strongly recommend running it on dedicated test machines that do not contain or provide access to sensitive information. Additionally, you should thoroughly review your lab network topology and access permissions before using this program.

Step	See...
Review the information about installing OpenText Functional Testing in an enterprise environment (if relevant)	"Enterprise deployment" on page 11
Ensure that you fulfill the installation prerequisites	"Installation prerequisites" below
Ensure that you have the required permissions for using OpenText Functional Testing	"Required access permissions" on page 23
Review the information about upgrading OpenText Functional Testing (if relevant)	"Upgrade" on page 53
Review any known issues	"Known issues when installing" on page 65
	"Known issues with licensing" on page 100
Review the instructions for using OpenText Functional Testing in a secure manner	OpenText Functional Testing Security Reference

Installation prerequisites

This section describes the prerequisites necessary to install OpenText Functional Testing. Before you install, verify the following prerequisites:

Prerequisite	Description
Install location	Select the location where you want to install OpenText Functional Testing. Do not install OpenText Functional Testing on a network drive. The installation path, and the path to the installation files, must include English characters only.
Computer state	Ensure that your computer does not need to be restarted.
System requirements	Ensure that your computer meets all minimum hardware and software requirements listed in the Support Matrix. As described in the Support Matrix, some of the prerequisites are included in the OpenText Functional Testing installation packages. You can install these as part of the OpenText Functional Testing installation process. Make sure AI Codeless Testing is **not** installed on your computer.
Licenses	Know which type of license you are using. If you are using a concurrent license, have the license server URL available. For more details, see "Licensing" on page 75.
Privileges	Make sure you have administrative privileges to install OpenText Functional Testing.
GUI testing add-ins	Know which add-ins you want to use for GUI testing. We recommend installing only those you expect to use.

Prerequisite	Description
API testing - SOAP activities	If you plan to run SOAP activities in your API tests, such as validating or creating checkpoints for SOAP messages, you must have the WS-I testing tools installed in the OpenText Functional Testing installation bin folder. To install these testing tools, perform the following steps: 1. From the official Web Services Interoperability Organization (WS-I) website, download the **C#** package of **Interoperability Testing Tools 1.1**. 2. Extract the contents of the file you downloaded. 3. Copy all of the content from the extracted **wsi-test-tools\cs\bin** folder to the **<OpenText Functional Testing installation>\bin** folder.
API testing - web services	If you plan to run web service testing using WSE security settings, you must have the .NET Framework 3.5, WSE 2.0sp3 package, and WSE 3.0 package installed on your computer. The .NET 3.5 Framework and WSE packages are not installed with the OpenText Functional Testing installation. If they are not installed on your computer, you can install them by following the instructions below: 1. Install and activate .NET 3.5 Framework. See the instructions in the MSDN: https://msdn.microsoft.com/en-us/library/hh506443(v=vs.110).aspx. 2. Download WSE 2.0 sp3 and WSE 3.0 packages from Marketplace: **https://marketplace.opentext.com/appdelivery/content/uft-one-installation-prerequisites** and install them. **Note:** In Windows 10 and later, Microsoft no longer officially supports WSE. For details on installing these prerequisites using the silent installation commands, see "Silent commands for installing prerequisites" on page 34. **Note:** If you use WCF for secure Web Service testing, these prerequisites are not necessary. For details on customizing security for WCF-type Web services, see the Help Center.

Required access permissions

Verify the following access permissions before you start running OpenText Functional Testing, or working with OpenText Functional Testing and OpenText Application Quality Management or BPT.

In this section:

- Required permissions for OpenText Functional Testing 23
- Required permissions for OpenText Application Quality Management 24
- Required permissions for BPT 25

Required permissions for OpenText Functional Testing

Required file system permissions

File system permissions	Description
Read/write permissions	You must have **read/write** permissions to the following files and folders, as well as any sub-folders: • The **Temp** folder • **User Profile** folders • The folder containing OpenText Functional Testing solutions, tests, or run results • The **%programdata%\OpenText** folder • The **%localappdata%\OpenText** folder • The **%appdata%\OpenText** folder
Read/execute permissions	The installation folder
Read permissions	You must have **read** permissions to the following folders: • The **Windows** folder • The **System** folder • The **Windows\System32** folder • The **<Windows>\mercury.ini** file

Required registry key permissions

Registry key permissions	Description
Read/write permissions	All keys under: - **HKEY_CURRENT_USER\Software\Mercury Interactive** or **[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Hewlett-Packard]** - **HKEY_CURRENT_USER\SOFTWARE\Hewlett-Packard**
Read and Query Value permissions	- **HKEY_LOCAL_MACHINE** keys - **HKEY_CLASSES_ROOT** keys

> **Note:** Some folder paths intentionally include previous company branding, due to backward compatibility considerations.

Required permissions for OpenText Application Quality Management

Permissions	Description
Read/write permissions	- ALM cache folder - **<Program Data>\Micro Focus** folder - OpenText Functional Testing Add-in for ALM installation folder
Administrative permissions	For the first connection

Required permissions for BPT

Ensure that you have the required ALM permissions before working with business components and application areas.

Component steps

To work with component steps, you must have the appropriate **Add Step**, **Modify Step**, or **Delete Step** permissions set.

You do not need **Modify Component** permission to work with component steps.

The **Modify Component** permission enables you to work with component properties, which are the fields in the component Details tab.

Parameters in OpenText Application Quality Management or other testing tools

To work with parameters, you must have all the parameter task permissions set in OpenText Application Quality Management.

Application areas

To modify application areas, you must have the separate permissions for resources required for modifying components, and adding, modifying, and deleting steps.

All four permissions (**Modify Component**, **Add Step**, **Modify Step**, or **Delete Step**) are required.

If one of these permissions is not assigned, you can open application areas only in read-only format.

Next steps:

- "Installation wizard" on the next page
- "Silent installation" on page 33
- "Install the Package for the Web Bundle" on page 48

Installation wizard

This section describes how to run the OpenText Functional Testing installation wizard, which guides you through the installation process. Before running this process, make sure you review the "Installation packages" on page 14 and the instructions in "Before you install" on page 20.

If you are installing on a computer using a language other than English, your installation setup and wizard automatically runs in the language of your computer.

In this section:

- Download the installation package 26
- Run the installation wizard 27
- Product improvement program 27
- Change the user interface language 28
- Custom Setup screen 28
- OpenText Functional Testing Configuration screen 30

Download the installation package

Access the installation packages from the free trial page or by selecting your account on the Software Licenses and Downloads page.

Download the Full UFT One DVD Release or Core UFT One DVD Bundle.

Extract the files from the downloaded **.zip** file and locate the **Setup.exe** file that runs the installation.

> **Caution:** Make sure that the installation files are extracted to an **empty** folder, at a location whose path does not exceed 80 characters.
>
> Windows is limited to 260 characters in these files. The installation will fail if the file paths of the individual installation files are longer. In such cases, move the extracted installation files to a location with a shorter path.

Run the installation wizard

Before the installation, restart your computer to ensure complete configuration of the system.

Then, run the **Setup.exe** file and select **OpenText Functional Testing Setup** from the installation start screen. Follow the steps as instructed to complete your installation.

For more details on the steps of the installation wizard, see below.

When the OpenText Functional Testing installation is complete, you are prompted to view the **Readme** and installation log.

You may also be prompted to restart your computer. If you are, we recommend restarting as soon as possible. Delaying the system restart may result in unexpected OpenText Functional Testing behavior.

> **Note:** If you are using Web 2.0 add-ins or extensibility toolkits, perform additional installations. For details, see "Install Web 2.0 add-ins or extensibility toolkits" on page 61.

Product improvement program

In the End-User License Agreement pane, you can select **Participate in the Functional Testing improvement program**.

If this option is selected, usage data is collected and sent back to OpenText. This data helps us understand which improvements matter most to users.

In the wizard, click **More Details** for details about the program.

> **Note:** You can later deactivate and reactivate this data collection inside the product configuration options. For more details, see Usage Data Collector pane in the Help Center.

Change the user interface language

By default, OpenText Functional Testing is installed in English.

If you want to install OpenText Functional Testing in the language of your operating system, select the language option at the bottom of the License Agreement screen.

Custom Setup screen

Select the features you want to install in the Custom Setup screen.

For each feature, select one of the following installation options:

- **Will be installed on local hard drive.**

  Installs the selected feature on your local hard drive. Sub-features of the selected feature are not installed.

- **Entire feature will be installed on local hard drive.**

  Installs the entire selected feature and its sub-features on your local hard drive.

  For example, you can install the .NET Add-in with its sub add-ins, such as Windows Presentation Foundation.

- **Entire feature will be unavailable.**

  Excludes the feature from the installation. It will not be available in OpenText Functional Testing.

  To install any of these features later, you can rerun the installation with the **Change** or **Modify** option and select them then.

The following table describes each feature listed:

Feature	Description
Runtime Engine	Mandatory. Enables you to run tests create d in OpenText Functional Testing or OpenText Functional Testing for Developers.

Feature	Description
UI Designer and IDE	Enables you to edit OpenText Functional Testing tests.
Run Results Viewer	Enables you to view test run results. Alternatively, you can view run results in a browser window without the Run Results Viewer.
Samples	Demo applications used in the tutorials.
ALM Plugin	Enables you to run and edit OpenText Functional Testing tests directly from OpenText Application Quality Management.
OpenText Functional Testing for Developers	Enables you to create functional tests directly from your developer IDE.
AI Object Detection	Enables you to use AI-based testing. (Requires a 64-bit operating system).

Feature	Description
ABBYY OCR Engine	Enables you to use ABBYY OCR text recognition. **Note:** • The ABBYY OCR Engine files are included in the Full UFT One DVD Release and in the Package for the Web Bundle. • If you already installed OpenText Functional Testing without the ABBYY OCR Engine and you want to include it in the installation now, you must run the installation program as an administrator. • The Core UFT One DVD Bundle does not include the ABBYY OCR Engine files. If you downloaded that installation package and then you want to install the ABBYY OCR Engine feature, do the following: a. Download the standalone UFT OCR Expansion Pack. Download the version that matches the Core UFT One DVD Bundle version you use. You can access the installation files from the free trial page or by selecting your account on the Software Licenses and Downloads page. b. As an administrator, run the installation program and select the ABBYY OCR Engine feature on the Custom Setup screen. c. On the next screen, specify the location of the UFT OCR Expansion Pack **.zip** file. • Installing this feature requires approximately 1GB of space.
GUI Testing Add-ins	Provide the support which enables you to test applications using supported technology versions. If you plan to test Web 2.0 applications, install the Web Add-in as well.

OpenText Functional Testing Configuration screen

Select any required items you want to configure automatically with the installation.

Configuration options include:

Option	Description
Set Chrome, Edge, Firefox options	Enables OpenText Functional Testing to automatically install the OpenText Functional Testing Agent extensions in Chrome, Edge, and Firefox. This option updates your browser policies and adds the extensions to your browsers' allowlist or installation list, enabling the extensions to be installed during OpenText Functional Testing installation. **Caution:** Make sure that your company's security policies permit you to make this change. Deselecting this option removes any browser policy updates performed by previous installations. If you deselect this option, you can rerun the installation later with the **Change** or **Modify** option and select it then. Alternatively, you can manually add the extensions to the lists of allowed extensions. For details on doing this manually, see the topics on web browser known issues in the OpenText Functional Testing Help Center.
Configure Internet Explorer settings	Enables OpenText Functional Testing to use the Microsoft Script Debugger application during test runs. Alternatively, configure these settings manually before running OpenText Functional Testing. In **Internet Options** > **Advanced**, select the following options: • **Disable script debugging** • **Enable third-party browser extensions**
Allow running OpenText Functional Testing remotely from OpenText Application Quality Management	Changes DCOM permissions and security settings and opens a firewall port on your OpenText Functional Testing computer. Required if you want to run tests remotely from OpenText Application Quality Management. To set these options later on, see the Additional Installation Requirements Utility topic in the Help Center.

Option	Description
Allow running OpenText Functional Testing remotely from Automation Scripts	Changes DCOM permissions and security settings to enable other computers to remotely control OpenText Functional Testing using automation scripts. To set these options later on, see the Additional Installation Requirements Utility topic in the Help Center.

> **Caution:** Running OpenText Functional Testing remotely from Automation Scripts enables remote users to control OpenText Functional Testing on this machine, exposing the machine to security risks.

Silent installation

You can install OpenText Functional Testing and the OpenText Application Quality Management Add-in silently on your computer or a remote computer.

To install the Package for the Web Bundle of OpenText Functional Testing silently, see "Install the Package for the Web Bundle" on page 48.

In this section:

- Before running a silent installation 33
- Silent commands for installing prerequisites 34
- Install OpenText Functional Testing silently 37
- Include specific features 38
- Set configuration options 43
- Additional command properties 45
- Install localized versions of OpenText Functional Testing 46
- Install Web 2.0 Add-ins 46
- Install a standalone OpenText Functional Testing Add-in for ALM 47

Before running a silent installation

Before installing silently:

- **Review** the "Installation packages" on page 14 and the information in "Before you install" on page 20.
- **Verify** that you have administrator privileges.
- **Save** any open files and close all open applications.
- **Restart** your system to ensure a complete system configuration.
- Note that silent installation commands are **case-sensitive** and should be entered exactly as listed.

Installation file location:

All of the silent installation commands below run programs from the **<OpenText Functional Testing Installation files>** folder.

These are the files available after you download an installation package and extract it:

- Full UFT One DVD Release or Core UFT One DVD Bundle: You downloaded a **.zip** file and extracted it.
- Package for the Web Bundle: You downloaded a **Setup.exe** file and ran it to extract the package content.

> **Caution:** Make sure that the installation files are extracted to an **empty** folder, at a location whose path does not exceed 80 characters.
>
> Windows is limited to 260 characters in these files. The installation will fail if the file paths of the individual installation files are longer. In such cases, move the extracted installation files to a location with a shorter path.

Silent commands for installing prerequisites

Use the command syntaxes listed below to install OpenText Functional Testing prerequisites.

If you are installing only the OpenText Functional Testing Add-in for ALM or the OpenText Functional Testing Run Results Viewer, install a subset of these prerequisites. For details, see "Install prerequisites for the OpenText Functional Testing Add-in for ALM or the OpenText Functional Testing Run Results Viewer" on page 37.

> **Note:**
>
> - Some items have different commands for different systems. Run the command most appropriate for your system.
> - In Windows 10 and later, Microsoft no longer officially supports WSE.

Install all OpenText Functional Testing prerequisites

```
<OpenText Functional Testing Installation files>\Unified Functional
Testing\EN\setup.exe /InstallOnlyPrerequisite /s
```

Install .NET Framework 4.8

```
<OpenText Functional Testing Installation
files>\prerequisites\dotnet48\ndp48-x86-x64-allos-enu.exe /q
/norestart
```

Install .NET Desktop Runtime (required for the UIA Pro add-in and the Parallel Runner UI)

Install .NET 8 Desktop Runtime using one of the following commands:

```
<OpenText Functional Testing Installation
files>\prerequisites\dotnet_desktop_runtime8_6\windowsdesktop-
runtime-8.0.6-win-x86.exe /q /norestart
```

```
<OpenText Functional Testing Installation
files>\prerequisites\dotnet_desktop_runtime8_6\windowsdesktop-
runtime-8.0.6-win-x64.exe /q /norestart
```

Install Microsoft Visual C++ 2022 Redistributable

Install Microsoft Visual C++ 2022 Redistributable using one of the following commands:

```
<OpenText Functional Testing Installation
files>\prerequisites\vc2022_redist_x86\vcredist_x86.exe /quiet
/norestart
```

```
<OpenText Functional Testing Installation
files>\prerequisites\vc2022_redist_x64\vcredist_x64.exe /quiet
```

```
/norestart
```

Install Microsoft Access Database Engine 2016

```
<OpenText Functional Testing Installation
files>\prerequisites\msade2016\AccessDatabaseEngine.exe /quiet
```

Install Microsoft WSE 2.0 SP3 Runtime (required only to run web service testing using WSE security settings)

Download the **.msi** from https://marketplace.opentext.com/appdelivery/content/uft-one-installation-prerequisites and run the following command.

```
MicrosoftWSE2.0SP3Runtime.msi /quiet /norestart ALLUSERS=1
```

Install Microsoft WSE 3.0 Runtime (required only to run web service testing using WSE security settings)

Download the **.msi** from https://marketplace.opentext.com/appdelivery/content/uft-one-installation-prerequisites and run the following command.

```
MicrosoftWSE3.0Runtime.msi /quiet /norestart ALLUSERS=1
```

Install Microsoft PDM installer

Use one of the following commands:

```
<OpenText Functional Testing Installation
files>\prerequisites\pdm\ScriptDebugging_x86.msi /quiet /norestart
```

```
<OpenText Functional Testing Installation
files>\prerequisites\pdm\ScriptDebugging_x64.msi /quiet /norestart
```

Install prerequisites for the OpenText Functional Testing Add-in for ALM or the OpenText Functional Testing Run Results Viewer

Install the following prerequisites on your machine when installing only the OpenText Functional Testing Add-in for ALM or the OpenText Functional Testing Run Results Viewer:

"Install .NET Framework 4.8" on page 35

Install OpenText Functional Testing silently

Run the msiexec command to install OpenText Functional Testing using the following syntax.

If you do not specify an installation folder, OpenText Functional Testing is installed in the default installation folder.

64-bit

```
msiexec /i "<OpenText Functional Testing Installation files>\Unified
Functional Testing\MSI\Unified_Functional_Testing_x64.msi" /qb
```

32-bit

```
msiexec /i "<OpenText Functional Testing Installation files>\Unified
Functional Testing\MSI\Unified_Functional_Testing_x86.msi" /qb
```

Default features and add-ins

The following features and add-ins are installed by default:

- Runtime Engine
- UI Designer and IDE
- Run Results Viewer
- Samples
- AI Object Detection
- ABBYY OCR Engine

> **Note:** If you downloaded the Core UFT One DVD Bundle, the ABBYY OCR Engine feature is not installed by default. For details, see "Installation packages" on page 14.

- Add-ins:
  - ActiveX Add-in
  - Visual Basic Add-in
  - Web Add-in

If you want to customize which features are installed, see "Include specific features" below.

Other msiexec options are supported as well.

To repair an existing installation, use:

```
msiexec /q /fa "<OpenText Functional Testing Installation
files>\Unified Functional Testing\MSI\Unified_Functional_Testing_
x<64 or 86>.msi"
```

To uninstall the program, use:

```
msiexec /q /x "<OpenText Functional Testing Installation
files>\Unified Functional Testing\MSI\Unified_Functional_Testing_
x<64 or 86>.msi"
```

Include specific features

By default, the silent installation command installs OpenText Functional Testing with default features and add-ins, as described in "Default features and add-ins" on the previous page.

To specify which features and add-ins to include in the installation, add the ADDLOCAL MSI property to the silent installation command. In the ADDLOCAL property's list of features, specify what to include in the installation by using the values described below.

Note:

- When using the ADDLOCAL property, you must include the **Core_Components** feature, which installs the Runtime Engine.
- Values must be separated by commas and must not contain spaces.
- When installing a feature using the ADDLOCAL property, its parent always is installed as well.

The following examples show the syntax of the ADDLOCAL property and its associated features.

This example uses the ADDLOCAL property to install the OpenText Functional Testing Runtime Engine only:

```
msiexec /i "<OpenText Functional Testing Installation files>\Unified
Functional Testing\MSI\Unified_Functional_Testing_x64.msi" /qb
ADDLOCAL="Core_Components" INSTALLDIR="<FolderName>" ALLOW_
OTHERSRUNTESTS=1
```

This example uses the ADDLOCAL property to perform a standard installation with the Java Add-in:

```
msiexec /i "<OpenText Functional Testing Installation files>\Unified
Functional Testing\MSI\Unified_Functional_Testing_x64.msi" /qb
ADDLOCAL="Core_Components,IDE,Test_Results_Viewer,Samples,Java_Add_
in" INSTALLDIR="<FolderName>">"
```

This example uses the ADDLOCAL property to perform a standard installation with Web and Java Add-ins and the DCOM configurations set:

```
msiexec /i "<OpenText Functional Testing Installation files>\Unified
Functional Testing\MSI\Unified_Functional_Testing_x64.msi" /qb
ADDLOCAL="Core_Components,Samples,Java_Add_in" CONF_DICOM=1
```

```
INSTALLDIR="<FolderName>"
```

For more details, see:

- "ADDLOCAL values for OpenText Functional Testing features" below
- "ADDLOCAL values for installing Add-ins" on the next page
- "ADDLOCAL values for OpenText Functional Testing for Developers components" on page 42

ADDLOCAL values for OpenText Functional Testing features

Value	Description
Core_ Components	(Mandatory) Installs the OpenText Functional Testing Runtime Engine.
IDE	Installs the OpenText Functional Testing user interface.
Test_Results_ Viewer	Installs the Run Results Viewer.
Samples	Installs the sample applications included with the OpenText Functional Testing installation.
ALM_Plugin	Installs the OpenText Functional Testing Add-in for ALM. **Note:** OpenText Functional Testing Add-in for ALM is installed with the same user interface language as OpenText Functional Testing.
AIServices	Enables you to use AI Object Detection and AI-based testing in OpenText Functional Testing. (Requires a 64-bit operating system).

Abbyy_OCR	Enables you to use ABBYY OCR text recognition in OpenText Functional Testing. **Note:** The Core UFT One DVD Bundle does not include the ABBYY OCR Engine files. If you downloaded that installation package and then you want to install the ABBYY OCR Engine feature, do the following: 1. Download the standalone UFT OCR Expansion Pack. Download the version that matches the Core UFT One DVD Bundle version you use. You can access the installation files from the free trial page or by selecting your account on the Software Licenses and Downloads page. 2. Add the UFT OCR Expansion Pack **.zip** file location to the msiexec command, using the following syntax: `msiexec /norestart /qn /i "<Installation files>\Unified Functional Testing\MSI\Unified_Functional_Testing_x64.msi" /l*xv "C:\InstallLogs.log" ADDLOCAL="Core_Components,Abbyy_OCR,Samples" ABBYY_SOURCEFILE="<UFT OCR Expansion Pack path>\UFT_<version number>_OCR_Expansion_Pack.zip"`

ADDLOCAL values for installing Add-ins

Use these values in the ADDLOCAL property's list of features to install various add-ins:

- **ActiveX_Add_in**
- **Visual_Basic_Add_in**
- **Web_Add_in**
- **Delphi_Add_in**
- **Java_Add_in**
- **_Net_Add_in**
- **WPF_Add_in**
- **Oracle_Add_in**

- **PDF_Add_in**
- **PeopleSoft_Add_in**
- **PowerBuilder_Add_in**
- **Qt_Add_in**
- **SAP_Solutions_Add_in**
- **SAP_eCATT_integration**
- **Siebel_Add_in**
- **Stingray_Add_in**
- **TE_Add_in**
- **VisualAge_Add_in**

If you plan to use the Web 2.0 Add-ins, you must install the Web Add-in as well. To install Web 2.0 Add-ins, see "Install Web 2.0 Add-ins" on page 46.

ADDLOCAL values for OpenText Functional Testing for Developers components

Value	Description
UFTDeveloper_Engine	Installs the run-time engine.
UFTDeveloper_Client	Installs the client.
Vs2017Addin **Vs2019Addin** **Vs2022Addin**	Installs the plugin for the relevant version of Microsoft Visual Studio.
IntelliJAddin	Installs the plugin for IntelliJ IDEA.
EclipseAddin	Installs the plugin for Eclipse.
ECLIPSE_INSTALLDIR	The path to your Eclipse IDE.

Set configuration options

This section lists the silent installation properties to use for the installation configuration options described in "OpenText Functional Testing Configuration screen" on page 30.

Configuration option	Property to add to the installation command
Set Chrome, Edge, Firefox options	**ALLOW_BROWSER_EXT** **Default:** 1. Run the installation with **ALLOW_BROWSER_EXT=0** to prevent OpenText Functional Testing from updating your browser policies or to undo updates performed by previous OpenText Functional Testing installations.
Configure Internet Explorer settings	**CONF_MSIE**
Allow running OpenText Functional Testing remotely from OpenText Application Quality Management	**ALLOW_RUN_FROM_ALM** **Default:** 0. To set this option for silent installation, set its value to 1.
Allow running OpenText Functional Testing remotely from Automation Scripts	**ALLOW_RUN_FROM_SCRIPTS** **Default:** 0. To set this option for silent installation, set its value to 1.
IMPROVEMENTPROGRAM=0	(Optional) Instructs the silent installation to deactivate usage data collection for the OpenText Functional Testing improvement program. For details, see "Product improvement program" on page 27.

Configuration option	Property to add to the installation command
WEB_EXTENSION_INSTALL_MODE	Enables you to control how the OpenText Functional Testing Agent browser extension is installed on your browser. For details on the installation modes, see the topics on setting up the OpenText Functional Testing Agent in theHelp Center. Specify the installation mode separately for Chrome and Edge in the following format: WEB_EXTENSION_INSTALL_MODE="chrome_setting;edge_setting" Possible values for chrome_setting: • InstallChromeStoreExtension_Normal • InstallChromeStoreExtension_Force • TempLoadLocalChromeExtension Possible values for edge_setting: • InstallEdgeStoreExtension_Normal • InstallEdgeStoreExtension_Force • TempLoadLocalEdgeExtension Note: To use this option, you must also set the **ALLOW_BROWSER_EXT** option to **1**. **Caution:** Make sure that your company's security policies permit you to use the option you select.

Caution: Running OpenText Functional Testing remotely from Automation Scripts enables remote users to control OpenText Functional Testing on this machine, exposing the machine to security risks.

By default, the silent installation does not configure DCOM settings required for remotely controlling OpenText Functional Testing using automation scripts.

To configure the DCOM settings for automation scripts, use the following syntax in your silent installation command:

```
ALLOW_RUN_FROM_ALM=1
ALLOW_RUN_FROM_SCRIPTS=1
```

Additional command properties

This section lists additional properties that you can use in the OpenText Functional Testing silent installation command:

Command / Argument	Description
LICID=<license ID>	(Optional) The license ID to specify when installing OpenText Functional Testing licenses. **Default:** 20528 (Functional Testing Concurrent User). **Note:** If you installed the OpenText Functional Testing for Developers feature when installing OpenText Functional Testing, using this feature requires license ID 10594 (UFT One Concurrent User) or 23078 (UFT Ultimate Concurrent User).
LICSVR=<server name>	(Mandatory if you provided a license ID) The name or IP address of the license server to specify when installing OpenText Functional Testing licenses.
MsiProperties	(Optional) Any MSI properties or parameters, such as **INSTALLDIR**. Each MSI property and its definition must be contained in quotes ("") and must not contain spaces. **Note:** You can use **INSTALLDIR** to specify the installation folder only when performing a new installation. When you run the silent installation in an upgrade scenario, OpenText Functional Testing is installed in the same location as the previous version.
MsiFlags	(Optional) Any MSI options, flags, and other instructions that are not included in the MsiProperties arguments, such as logging commands.

Install localized versions of OpenText Functional Testing

By default, OpenText Functional Testing is installed in English.

To install OpenText Functional Testing in the language of your operating system, add the PRODUCT_LOCALE property to the **msiexec** command. The property value you use determines the language of the localized OpenText Functional Testing.

Make sure to use the property value that matches your operating system language, otherwise OpenText Functional Testing is installed in English.

OS Language	Value for the PRODUCT_LOCALE property
Chinese	**"CHS"**
French	**"FRA"**
German	**"DEU"**
Japanese	**"JPN"**

The following example installs the Chinese version of OpenText Functional Testing and uses the ADDLOCAL property to install the .NET Add-in:

```
 msiexec /i "<OpenText Functional Testing Installation
files>\Unified Functional Testing\MSI\Unified_Functional_Testing_
x64.msi" /qb ADDLOCAL="Core_Components,Samples,_Net_Add_in" PRODUCT_
LOCALE="CHS" INSTALLDIR="FolderName>"
```

Install Web 2.0 Add-ins

OpenText Functional Testing's Web 2.0 Add-ins, such as JQueryUI or Dojo, are supported as part of OpenText Functional Testing's Web extensibility, which requires a separate installation.

Install Web 2.0 Add-ins using an msiexec command in the following syntax:

```
msiexec /qn /i "<OpenText Functional Testing Installation
files>\Extensibility and Toolkits\Web2AddinSetup\Web2AddinSetup.msi"
ADDLOCAL=ASPAjax,Dojo,GWT,jQueryUI,YahooUI,SiebelOpenUI,ExtJS,Sales
forceLightning
```

Include or exclude the specific ADDLOCAL values for the Web 2.0 Add-ins you need.

Install a standalone OpenText Functional Testing Add-in for ALM

To install the OpenText Functional Testing Add-in for ALM without installing OpenText Functional Testing, run the standalone ALM plugin MSI file.

In the command line, run the msiexec command to install the OpenText Functional Testing Add-in for ALM, using the following syntax:

```
msiexec /i "<OpenText Functional Testing Installation
files>\ALMPlugin\MSI\Unified_Functional_Add-in_for_ALM.msi" /qn
```

You can install the OpenText Functional Testing Add-in for ALM in the language of your operating system by adding the PRODUCT_LOCALE property to the msiexec command. For details on the PRODUCT_LOCALE property, see "Install localized versions of OpenText Functional Testing" on the previous page.

Next steps:

- "Verify the installation" on page 59

Install the Package for the Web Bundle

This section explains how to install OpenText Functional Testing from the lighter OpenText Functional Testing installation package. Before running this process, make sure you review the instructions in "Before you install" on page 20.

In this section:

- Overview 48
- Download the Package for the Web Bundle 49
- Install OpenText Functional Testing using an installation wizard 49
- Run installation from a command line interface 50

Overview

The Package for the Web Bundle is a self-extracting installation package, which contains an MSI installation program.

Installing this version of OpenText Functional Testing includes the following steps:

1. Extract the installation package.
2. Run the MSI installation program to install OpenText Functional Testing.

You can run these steps together automatically, or you can perform them separately.

This package includes the OpenText Functional Testing prerequisites, but does not install them automatically. If you need to install any prerequisites, run their installation programs after you extract the installation package and before you run the MSI installation.

The sections below describe running this installation using a UI wizard or a command-line interface.

For details on other OpenText Functional Testing installation packages, see "Installation packages" on page 14.

> **Caution:** Make sure that the installation files are extracted to an **empty** folder, at a location whose path does not exceed 80 characters.
>
> Windows is limited to 260 characters in these files. The installation will fail if the file paths of the individual installation files are longer. In such cases, move the extracted installation files to a location with a shorter path.

Download the Package for the Web Bundle

Access the installation packages from the free trial page or by selecting your account on the Software Licenses and Downloads page.

Download the OpenText Functional Testing Package for the Web Bundle.

Install OpenText Functional Testing using an installation wizard

Run the ***Setup.exe** file you downloaded. This file self-extracts, providing the necessary installation files, and then automatically performs the installation.

If you need to install any of the prerequisites provided in this package:

1. Stop the wizard after it completes the file extraction.
2. Install all missing prerequisites by running the **setup.exe** file included in the extracted files.

   Alternatively, install specific prerequisites manually using the executable files available in the **prerequisites** folder included in the extracted files.
3. Run the **.msi** file included in the extracted files to performs the OpenText Functional Testing installation.

For details on the selections and configuration details you can customize during the installation, see "Installation wizard" on page 26.

Run installation from a command line interface

When using a command line to run the ***Setup.exe** file you downloaded, you can control various aspects of the installation process:

- You can choose whether to install silently or to display a user interface.
- You can specify whether to extract the installation package and automatically run the MSI program, or to stop after the extraction.

  Stopping the process after the file extraction enables you to:

  - Install prerequisites before running the installation.
  - Specify command line options for the MSI run, instructing it to run silently, or configuring features, add-ins, and options to customize the installation.

Use the following commands to install OpenText Functional Testing:

Command Syntax	Description
UFT_One_<version>_ Setup.exe -y	Extracts the installation package and installs OpenText Functional Testing using a simple UI (a single dialog box with a progress bar only).
UFT_One_<version>_ Setup.exe -y -gm2	Extracts the installation package silently, and installs OpenText Functional Testing using the full installation wizard UI.
UFT_One_<version>_ Setup.exe -InstallPath="c:\<path>"	Extracts the installation package to a specific target folder instead of the default.

Command Syntax	Description
UFT_One_<version>_ Setup.exe -! <parameter list>	Extracts the installation package and installs OpenText Functional Testing, passing the defined parameter values to the MSI installer. Possible parameters: • **/s.** Do not present the prerequisite dialog box. Without this option, user intervention is required during installation to close this dialog box. • **/qn.** Install OpenText Functional Testing completely silently. • **/l*v "<custom log file location>.log".** Generate the installation log file in a custom location; the default location is **%temp%**. • **INSTALLDIR="<installation folder>".** Install OpenText Functional Testing to a custom location (not relevant in upgrade scenario). • **ADDLOCAL="<features to install>".** For details, see "Include specific features" on page 38.
UFT_One_<version>_ Setup.exe - ExecuteFile=""	Extracts the installation package without running the OpenText Functional Testing installation. When the extraction is complete, you can install prerequisites, as described in "Silent commands for installing prerequisites" on page 34. You can also use the commands and options described in "Silent installation" on page 33, to perform a silent OpenText Functional Testing installation, selecting the required add-ins. **Tip:** In the silent installation commands, change **<OpenText Functional Testing Installation files>** to the folder where you extracted the installation content.

Examples:

Extract the package to the default location (**c:\temp**), but do not start the installation:

```
UFT_One_2021_Setup.exe -y -ExecuteFile=""
```

Extract the package to the specified location (**c:\UFTinstall**) in silent mode, but do not start the installation:

```
UFT_One_2021_Setup.exe -y -gm2 -InstallPath="c:\UFTinstall" -
ExecuteFile=""
```

Extract the package silently and start installing OpenText Functional Testing using the simple UI:

```
UFT_One_2021_Setup.exe -y
```

Extract the package and start installing OpenText Functional Testing silently:

```
UFT_One_2021_Setup.exe -y -gm2 -! /s /qn
```

Extract the package to a specified location, install OpenText Functional Testing silently, customizing the log file location and installation location, and installing several add-ins and features:

```
UFT_One_2021_Setup.exe -InstallPath="C:\2021\extractedUFT" -y -gm2
-! /s /qn /l*v "C:\UFT_INSTALL.log" INSTALLDIR="C:\UFT_Program"
ALLOW_RUN_FROM_ALM=1 ALLOW_RUN_FROM_SCRIPTS=1 CONF_MSIE=1 DLWN_
SCRIPT_DBGR=1 ADDLOCAL="Core_Components,Web_Add_in,ALM_Plugin,Test_
Results_Viewer,Samples,ActiveX_Add_in,Visual_Basic_Add_in,Delphi_
Add_in,Flex_Add_in"
```

Next steps:

- "Verify the installation" on page 59

Upgrade

You can upgrade directly to the most recent version of OpenText Functional Testing from any previous version of the product.

In this section:

- Before you upgrade 53
- Perform your upgrade 53
- Configuration file location 54
- Notes for upgrading 55

Before you upgrade

Upgrading to the latest version ensures that you benefit from all the latest features and developments, including fixes and security updates. For more details, see the version upgrade hub.

Upgrading from a version earlier than 26.1 This version introduces enhancements to the UIA Pro Add-in, including some OMR changes. If you didn't enable the Enhanced Beta UIA Pro Add-in in the previous release, you might need to update your existing UIA Pro tests after upgrading. For details, see Object Model changes introduced in version 26.1.

If you are upgrading from version 2021 or earlier, consider the following:

If you customized the **EmulatedDevices.xml** file, back up the file before you upgrade. When you complete the upgrade, add your customization to the new **EmulatedDevices.xml** file located in the **<OpenText Functional Testing installation folder/bin>** directory.

Perform your upgrade

To upgrade OpenText Functional Testing:

1. Download the installation files for your new version from the free trial page or by selecting your account on the Software Licenses and Downloads page.

2. Restart your system to ensure complete configuration for the system.
3. Run the **OpenText Functional Testing_<VersionNumber>_Setup.exe** file to upgrade using the installation wizard.

   Alternatively, update your silent installation scripts to use the newly downloaded files.

> **Note:** Upgrades retain **Run Session** and **Startup** options only. Redefine all other settings as needed.

Configuration file location

Starting from OpenText Functional Testing 24.2, the configuration files are stored in new locations.

The first time you run OpenText Functional Testing after upgrading from a version earlier than 24.2, all existing setting files are automatically migrated to the new locations. This enables you to continue using your existing product settings in the new version.

If the Settings Migration fails to migrate all files, you are prompted to move the files manually. Otherwise, default settings are used instead of your existing configuration.

The following table lists the folders that contain files to migrate. Copy all of the files from the folders used in your previous version to the folders listed for your current version.

Versions 24.2 or 24.4	Versions 2021-23.4	Versions earlier than 2021
%APPDATA%\OpenText\UFT	%APPDATA%\Micro Focus\UFT	%APPDATA%\Hewlett-Packard\UFT

Versions 24.2 or 24.4	Versions 2021-23.4	Versions earlier than 2021
%APPDATA%\OpenText\QuickTest Professional	%APPDATA%\Micro Focus\QuickTest Professional	%APPDATA%\HP\Quick Test Professional
%APPDATA%\OpenText\API Testing	%APPDATA%\Micro Focus\API Testing	%APPDATA%\HP\API Testing
%PROGRAMDATA%\OpenText\UFT	%PROGRAMDATA%\Micro Focus\UFT	%PROGRAMDATA%\Hewlett-Packard\UFT
%LOCALAPPDATA%\OpenText\UFT	%LOCALAPPDATA%\Micro Focus\UFT	%LOCALAPPDATA%\HP\UFT

Note: If you downgrade as follows, the existing configuration will not be preserved and default settings of the older version are used:

Downgrade from 24.2 or later to 23.4 or earlier.

Downgrade from 2021 or later to 15.0.2 or earlier.

Notes for upgrading

The following items address issues when upgrading in specific situations. Read the instructions for any of the situations that are relevant for your upgrade.

- "Upgrade silent installation scripts" on the next page
- "Upgrade licenses" on the next page
- "Upgrade with text recognition options in automation scripts" on the next page
- "Upgrade" on page 53
- "Upgrade from a version earlier than 2023" on page 57
- "Upgrade from version 23.4 or earlier" on page 57
- "Upgrade from a version earlier than 24.2" on page 57
- "Upgrade from a version earlier than 25.2" on page 58

Upgrade silent installation scripts

If you are upgrading your silent installation scripts and the current scripts, including the **Help_Documents** parameter, remove this parameter. Help documents are no longer installed with OpenText Functional Testing.

You can access the Help Center online, or download it to your local drive. Download the Help from the Options dialog box (**Tools** > **Options** > **General** tab > **Help**).

Upgrade licenses

You can upgrade your license to the new Functional Testing licenses: UFT One and UFT Developer. This step is not required.

For assistance, contact your sales representative.

Upgrade with text recognition options in automation scripts

If you are running OpenText Functional Testing with an automation script and have added text recognition options to the script, the following properties are obsolete and must be updated:

Update	To
TextRecognitionLanguages	AbbyyOcrLanguages
TextRecognitionOrder	TextRecognitionOcrMechanism

Upgrade from a version earlier than 2021

The first time you run OpenText Functional Testing after upgrading from a version earlier to 2021, all existing setting files are automatically migrated to new locations.

If you install a different license immediately after the upgrade, the license is automatically overwritten the first time you run OpenText Functional Testing, resulting in failure.

Therefore, after the upgrade, open OpenText Functional Testing once before installing a new license.

Upgrade from a version earlier than 2023

In OpenText Functional Testing, identifying AI objects using ordinal location is calculated more consistently. As a result, you may need to adjust some steps in your existing tests.

If you find that AI objects described by ordinal location are not identified correctly, reinspect your application to find the best description to use. For example, if you described the object as third from the top, a description of fourth from the left might now give more consistent results.

For details on using ordinal locations for AI objects, see the OpenText Functional Testing Help Center.

Upgrade from version 23.4 or earlier

Some text recognition enhancements were introduced in versions 23.4 and 24.2. If you run tests last updated in earlier versions, you may need to adjust some text recognition settings:

- Changed in 23.4: AI-based text recognition

  Options for Noise trimming and considering UI control borders were added (selected by default). For details on configuring AI text recognition settings, see the OpenText Functional Testing Help Center.

- Changed in 23.4 and 24.2: Text recognition using the ABBYY OCR engine

  OpenText Functional Testing now uses a newer version of the ABBYY OCR engine. If necessary, adjust your text recognition settings. You can use the **Preview** pane to experiment with various settings until you achieve optimal recognition. For details on ABBYY text recognition settings, see the OpenText Functional Testing Help Center.

Upgrade from a version earlier than 24.2

If you are upgrading from a version earlier than 24.2, consider the following issues:

- If you previously installed the Core UFT One DVD Bundle and the UFT OCR Expansion Pack, make sure you download the new version of both installation packages when you upgrade. Core UFT One DVD Bundle 24.2 requires UFT OCR Expansion Pack 24.2 to support the ABBYY OCR engine.

- In 24.2, SAP SuccessFactors object identification was modified. If any of your SAP or Web objects in existing SAP SuccessFactors tests are not identified, update your tests by doing one of the following:
  - In the Object Repository, update the objects' descriptions using the **Update from Application** button. For details, see the OpenText Functional Testing Help Center.
  - Learn the object again by spying on it and adding it to the Object Repository. Then update the test steps to use the new object.
  - Delete the relevant steps and recreate them by recording.

Upgrade from a version earlier than 25.2

In 24.4 and earlier, after importing data from an Excel file using the **DataTable.Import** method, **GetCurrentRow** always returned 1.

In 25.2 and later, **GetCurrentRow** now correctly returns the number of the data table row being used for the current iteration.

Check tests written in previous versions to adjust for this change in behavior, if necessary.

Verify the installation

Use the OpenText Functional Testing Installation Check Tool to verify the status of your installation.

In this section:

- Run the Installation Check Tool analysis 59
- Learn an Installation Check Tool report 59

Run the Installation Check Tool analysis

After installation, access the Installation Validation tool as follows:

1. Open the **Additional Installation Requirements** utility (available from the Windows Start menu).
2. Click **Run** to run the utility, which handles any configuration prerequisites for working with OpenText Functional Testing. Run through any wizards as needed for your installation.
3. Open the **OpenText Installation Check Tool** (available from the Windows Start menu).
4. In the Installation Check Tool dialog, click **Analyze** to generate a report on the current installation and configuration status.
5. When the report is generated, click one of the following, as needed:
   - **View Report.** View the report as an htm file in a browser.
   - **Send Email.** Send the report to another user. You must have a default email application configured on the machine to use this option.

Learn an Installation Check Tool report

The Installation Check tool validates the installation and configuration state against their expected values.

Values that are returned as expected are highlighted in green, and unexpected values are highlighted in red.

> **Note:**
>
> - The Installation Check Tool returns the data on Remote Agent Settings dialog only if the Remote Agent is run in administrator mode.
> - The tool shows the permissions available on various folders and registration keys that might be relevant to OpenText Functional Testing's functionality. This information may be useful for the Support team if you require assistance.

See also:

- "Upgrade" on page 53
- "Installation prerequisites" on page 20
- "Known issues when installing" on page 65

Install Web 2.0 add-ins or extensibility toolkits

This section explains how to install Web 2.0 add-ins or extensibility toolkits.

- Web 2.0 add-ins enable you to test HTML user-interface objects (controls) in Web 2.0 environments. For a list of available Web 2.0 add-ins, see Web 2.0 Add-ins in the Help Center.
- Extensibility toolkits enable you to develop support for test objects not currently supported by the existing add-ins.

> **Note:** This installation is available only if you installed OpenText Functional Testing from the Full UFT One DVD Release or the Core UFT One DVD Bundle. For details, see "Installation packages" on page 14.

To install Web 2.0 add-ins or extensibility toolkits

1. Run the **Setup.exe** file and select the **Add-in Extensibility and Web 2.0 Toolkits** option from the installation start screen.
2. In the **Add-in Extensibility and Web 2.0 Toolkit support** page, select the Extensibility SDK or Web 2.0 Toolkits installation option that you want.
3. Follow the steps in the wizard to complete your installation.

After your installation, the toolkit files and Extensibility SDKs are found in the **Installdir\dat\Extensibility** folder.

Web 2.0 add-ins are displayed as child add-ins to the Web Add-in in the Add-in Manager when starting OpenText Functional Testing.

See also:

- "Installation packages" on page 14
- "Installation wizard" on page 26

Before connecting to OpenText Application Quality Management

If you intend to run OpenText Functional Testing tests remotely from OpenText Application Quality Management on this computer, you must modify User Account Control (UAC) settings before connecting. You can revert these modifications afterward.

> **Note:** The security changes described in this section should be performed by your System Administrator.
>
> Contact Microsoft Support if you have questions regarding changes in User Account Control (UAC) on any of these operating systems.

In this section:

- For Microsoft Windows 10 & 11, Windows Server 2016 & 2019 62
- For Microsoft Server 2012 63
- Enable UAC again if required 63

For Microsoft Windows 10 & 11, Windows Server 2016 & 2019

Modify UAC settings on Windows 10, Windows Server 2016, & Windows Server 2019 machines as follows:

1. Open the registry editor. (Run a **regedit** command)
2. Navigate to the following key: **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System**
3. Modify the **EnableLUA** DWORD value and set it to 0.
4. Restart the computer for your changes to take effect.

For Microsoft Server 2012

Modify UAC settings on Windows Server 2012 machines as follows:

1. Log in as an administrator.
2. From the Control Panel, select **User Accounts and Family Safety > User Accounts > Change User Account Control Settings**.
3. In the User Account Control Settings window, move the slider to **Never notify**.
4. In the Control Panel, select **System and Security > Administrative Tools > Local Security Policy**.
5. In the Local Security Policy window, in the left pane, select **Local Policies**.
6. In the Local Policies tree, select **Security Options**.
7. In the right pane, select the **User Account Control: Run all administrators in Admin Approval mode** option.
8. Select **Action > Properties** from the menu bar.
9. In the dialog that opens, select **Disabled**.
10. Restart the computer for your changes to take effect.

Enable UAC again if required

After connecting to OpenText Application Quality Management, return to the User Account Control Settings window to enable UAC again. Restore the slider to its previous position to turn the UAC option on again.

On Windows 10, open the registry editor and set the value of **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\EnableLUA** back to 1.

Restart the computer for your changes to take effect.

See also:

- Application Lifecycle Management
- OpenText Application Quality Management Help Center

Known issues when installing

This section describes troubleshooting and limitations for installing OpenText Functional Testing.

In this section:

- Files in use 65
- Component registration failed 66
- Change or repair the OpenText Functional Testing installation 66
- OpenText Functional Testing installations and other ADM software 66
- OpenText Functional Testing installations and Microsoft software 68
- Installing the OpenText Functional Testing Agent browser extension 71
- OpenText Functional Testing installations and 64-bit applications 72
- OpenText Functional Testing installations and Java 72
- Installing in languages other than English 73

Files in use

If the OpenText Functional Testing Files in Use dialog box is displayed during the installation process, select **Close the application and attempt to restart them.**

OpenText Functional Testing automatically closes the applications and continues the installation.

If after restarting, the OpenText Functional Testing Files in Use dialog box lists **Explorer** as the open application, do one of the following:

Option	Description
Close the applications and attempt to restart them	Instructs OpenText Functional Testing to automatically close the applications that are needed for the installation.
Do not close application	Instructs OpenText Functional Testing to continue the installation. You must restart your computer after the installation if you select this option.

Component registration failed

If a message is displayed during installation about component registration failing, do **not** click **OK** to continue.

Instead, check the issue in the **VC2015Prerequisite_yyyymmdd_XXXXXX.log** file in the **%TEMP%** directory. If the log shows that a service did not start correctly, restart the service manually, and start your installation again.

Change or repair the OpenText Functional Testing installation

Changing or repairing the OpenText Functional Testing installation requires write permissions to certain registry keys.

Repairing the installation from the control panel without these permissions behaves as follows:

The installation wizard stops and displays an error message.

- Log in to Windows as an administrator and **Change** or **Repair** the OpenText Functional Testing installation from the Control Panel.
- Run the OpenText Functional Testing MSI installation program from the OpenText Functional Testing installation package and select the **Change** or **Repair** option.

OpenText Functional Testing installations and other ADM software

The following describes troubleshooting and limitations for installing other ADM software with OpenText Functional Testing.

Option	Description
OpenText Sprinter	If you are using both OpenText Functional Testing and Sprinter on the same computer, and you modify either OpenText Functional Testing or Sprinter, you must run a **Repair** for the installation of the other product.
OpenText Application Quality Management	When an ALM client is installed on the same computer as OpenText Functional Testing, uninstalling OpenText Functional Testing may remove the association of video (**.fbr**) files. This may prevent you from viewing videos associated with defects in OpenText Application Quality Management using the Micro Player application. **Workaround:** Re-associate the video files with the Micro Player application from the Windows File Options dialog box.
OpenText Functional Testing for Developers	• In the **Custom Setup** screen, it is possible to select the OpenText Functional Testing for Developers Visual Studio or Eclipse plugins even if you do not have the relevant IDE installed at the time of the installation. If you install the IDE later, you will not have the OpenText Functional Testing for Developers plugin available. **Workaround**: After installing the required IDE, run a Repair for the installation. • Make sure to use the new syntax when silently installing OpenText Functional Testing for Developers as part of the OpenText Functional Testing installation. For details, see "ADDLOCAL values for OpenText Functional Testing for Developers components" on page 42. If you encounter an error similar to this one, make sure you did not use the old LeanFT silent installation command syntax: `Error: The installer has encountered an unexpected error installing this package. This may indicate a problem with this package. The error code is 2711. The arguments are: LeanFT`

OpenText Functional Testing installations and Microsoft software

The following describes the troubleshooting and limitation for installing OpenText Functional Testing on Windows or using OpenText Functional Testing with other Microsoft software.

Software	OpenText Functional Testing instructions
Windows 10 and 11	• When installing OpenText Functional Testing on a Windows 10 or 11 operating system, you must close Cortana and the Action Center before performing the OpenText Functional Testing installation. • You must have administrative permissions to connect to OpenText Application Quality Management from OpenText Functional Testing on Windows 10 or 11. Connect to ALM with Administrator permissions immediately after installing OpenText Functional Testing.

Software	OpenText Functional Testing instructions
pdm.dll	**To debug GUI tests in OpenText Functional Testing**, ensure that you have the **pdm.dll** file installed and registered. The **pdm.dll** file is installed and registered with Microsoft Visual Studio and Microsoft Office, and installed (but not registered) with Microsoft Internet Explorer. Alternatively, install the Microsoft Script Debugger provided with the OpenText Functional Testing installation, which provides the **pdm.dll**. **To register the pdm.dll file installed with Microsoft Internet Explorer:** 1. Ensure that you have administrator privileges. 2. Locate the **pdm.dll** file, usually located either in **c:\program files(x86)\internet explorer\** or **c:\program files\internet explorer**. 3. Move the **pdm.dll** file and the **msdbg2.dll** file from the same folder to a different location. 4. Run the following commands: **regsvr32 <full path to pdm.dll>\pdm.dll** **regsvr32 <full path to pdm.dll>\msdbg2.dll** **If your currently registered pdm.dll version is lower than 9:** 1. Uninstall the Microsoft Script Debugger (if installed). 2. Install the Microsoft Script Debugger using the OpenText Functional Testing Additional Installation Requirements utility. Open the **OpenText Functional Testing Additional Installation Requirements** utility from the Windows Start menu or by running **<Installdir>\bin\UFTInstallReqs.exe**.

Software	OpenText Functional Testing instructions

Microsoft Office 64-bit	Using OpenText Functional Testing and Microsoft Office 64-bit on the same machine requires some intervention. This is due to a conflict between the Microsoft Access Database Engine versions used by these two programs.
	Do one of the following:
	- Install OpenText Functional Testing before you install Microsoft Office 64-bit.
	- Before you install OpenText Functional Testing, install Microsoft Access Database Engine 2016 32-bit from an administrator command line, using the **/quiet** argument. For details, see "Install Microsoft Access Database Engine 2016" on page 36.
	- Install OpenText Functional Testing with the **SKIP_MSADE_CHECK** flag. This installs OpenText Functional Testing without requiring the Microsoft Access Database Engine 2016 Redistributable to be installed.
	Silently:
	`cmd /c MsiExec /norestart /qn /i "<OpenText Functional Testing MSI Path>\Unified_Functional_Testing_x64.msi" /l*xv "<log file path>.log" SKIP_MSADE_CHECK=1`
	Using the installation wizard:
	`cmd /c MsiExec /norestart /qf /i "<OpenText Functional Testing MSI Path>\Unified_Functional_Testing_x64.msi" /l*xv "<log file path>.log" SKIP_MSADE_CHECK=1`
	You can install the redistributable engine as described above, if you find later that you need it for API database activities.

Windows Server 2012 R2	When using OpenText Functional Testing on Windows Server 2012 R2, if you want to use API tests and components, make sure that you have MSU (Microsoft Update) KB2887595 installed.

Installing the OpenText Functional Testing Agent browser extension

The following provides instructions on installing OpenText Functional Testing Agent extension in browsers.

Browser	OpenText Functional Testing instructions
Google Chrome	If you are testing applications in Google Chrome version 68 or later, Chrome automatically downloads and installs the OpenText Functional Testing Agent for Google Chrome the first time you open Chrome after OpenText Functional Testing is installed. In the following cases, you must manually install the OpenText Functional Testing Agent Chrome extension: • You have no Internet connection. • You have not enabled the automatic updates for Google Chrome. • You are using Google Chrome version 67 or earlier. • You are using Google Chrome version 95 or earlier. For details on manually installing the extension, see the Help Center.
Mozilla Firefox	The first time you open Firefox after OpenText Functional Testing is installed, accept the prompt to install the OpenText Functional Testing Agent for Firefox.
General	To use the latest OpenText Functional Testing Agent browser extension, make sure none of its older versions are not installed. If both extensions are installed, manually remove the older before enabling the new.

OpenText Functional Testing installations and 64-bit applications

Option	Description
Installing with administrator privileges	OpenText Functional Testing loses support for 64-bit applications if a user with administrator privileges installs the OpenText Functional Testing Add-in for ALM, or performs a Repair operation on the Run Results Viewer, and then a user runs OpenText Functional Testing on that same computer without administrator privileges. **Workaround:** Log in as an administrator and do one of the following: • Repair OpenText Functional Testing • Run **<Installdir>\bin64\Mediator64.exe**.
32-bit and 64-bit applications	If your computer has two versions of an application, where one is 32-bit and the other is 64-bit, OpenText Functional Testing always opens the 32-bit version. This occurs when the operating system performs a redirect from the Program Files folder to the Program Files (x86) folder, and from the System32 folder to the SysWow64 folder. **Workaround**: To specify a 64-bit version, make sure that the step explicitly states the path to the 64-bit application.
.NET / WPF Add-in extensibility	When working with .NET or WPF Add-in extensibility for a 64-bit Windows Forms process, the custom server DLLs must be built using the **Any CPU** option.

OpenText Functional Testing installations and Java

When re-installing or upgrading the JRE on a machine with OpenText Functional Testing installed, you might encounter error 1603 preventing the JRE installation to complete.

This can be caused by an interference between the OpenText Functional Testing Java environment variables and the Java installer.

To successfully complete the installation, rename the OpenText Functional Testing Java environment variables, perform the JRE installation and restore the variable names.

To temporarily rename the OpenText Functional Testing Java environment variables:

1. On your Windows Desktop, right-click **My Computer** or **This PC** and select **Properties**.
2. Select the **Advanced** tab.
3. Click the **Environment Variables...** button.
4. Look for the following environment variables both under the user variable list and the system variables list and edit their names:
   - _JAVA_OPTIONS
   - Java_Tool_Options
   - IBM_Java_Options
5. Install the JRE.
6. After the installation is completed, change the environment variables names back to their original names.

Installing in languages other than English

When installing OpenText Functional Testing in languages other than English, the TTF16.ocx file is not registered by default. To avoid errors in these cases, do the following before beginning your installation:

1. Browse to the Windows Welcome screen and new user account settings. You may be able to find this in your Windows **Region** or **Region and Language** control panel settings.
2. Click **Copy settings...** and select to copy your current settings to the **Welcome screen and system accounts**.

See also:

- "Known issues with licensing" on page 100

Licensing

Working with OpenText Functional Testing requires a license. This section describes the different types of OpenText Functional Testing licenses, where to view your license information, and how to install licenses.

In this section:

- License types 75
- View license information 76
- AutoPass License Server 76
- Seat vs. concurrent licenses 77
- License editions 79
- Manage licenses with the wizard 82
- Manage licenses with the command line 89
- Configure license behavior 92
- Licensing FAQs 96
- Known issues with licensing 100

License types

When you install OpenText Functional Testing, you select one of the following license types:

- A permanent **seat** license that is specific to the computer on which it is installed.

  Note: The trial installation of OpenText Functional Testing includes a 30-day demo seat license.

- A network-based **concurrent** license that can be used by multiple OpenText Functional Testing users, taken from and returned to a licensing server pool.

  If you need a demo concurrent license, contact your OpenText Functional Testing sales representative or partner.

As long as you are logged in with administrator permissions, you can change your license type at any time. For example, if you are currently working with a seat license, you can choose to connect to a concurrent license server, if one is available on your network.

> **Note:** You can also open OpenText Functional Testing using a legacy QuickTest Professional license, but the functionality will be limited to GUI testing functionality.

For more details, see "Seat vs. concurrent licenses" on the next page and "License editions" on page 79.

For details about installing and configuring licenses, see:

- "Manage licenses with the wizard" on page 82
- "Manage licenses with the command line" on page 89
- "Configure license behavior" on page 92

View license information

To view details about your current license:

1. In OpenText Functional Testing, select **Help > About OpenText Functional Testing**.
2. Click **License**.

   If at least one of your licenses is about to expire, OpenText Functional Testing displays the date of the license closest to expiration.

AutoPass License Server

Concurrent licenses require the use of the AutoPass License Server. For supported versions of AutoPass, see the Support Matrix.

Download the installation from the Marketplace (requires login).

For more details, such as proxy settings or managing licenses and users, see the AutoPass License Server online documentation.

See also:

- "Known issues when installing" on page 65
- "Licensing FAQs" on page 96
- "Known issues with licensing" on page 100

Seat vs. concurrent licenses

This section describes OpenText Functional Testing seat and concurrent licenses, and can help you select the best type of license for your needs.

In this section:

- Seat licenses 77
- Concurrent licenses 78

Seat licenses

Seat licenses are machine-specific licenses, based on a specific locking code per computer.

The key must be entered once only, and provides one installation per key.

A computer with multiple bootable partitions may generate a different locking code for each partition. When obtaining a seat license key, you must use the locking code for the partition on which you will be using OpenText Functional Testing or OpenText Functional Testing for Developers.

Seat licenses and Windows servers

If you install a seat license on a Windows server, the seat license will be consumed by the first user who logs into the Windows server.

Limited seat licenses

If you install a time-limited seat license, do not modify the date on your computer. Doing so will block your active seat license and prevent future seat license installations on that computer. For more information, contact your OpenText Functional Testing license supplier.

MAC address or host name changes

If you modify the MAC address or host name of the computer after installing a seat license, you must regenerate and install your seat licenses again.

Concurrent licenses

Concurrent licenses are taken from the AutoPass License Server on a per-session basis. You must have an active network connection to install and access concurrent licenses.

Each time OpenText Functional Testing or OpenText Functional Testing for Developers starts, it tries to connect to the AutoPass License Server for an available license, which regulates the number of licenses currently in use.

The license is returned to the AutoPass License Server when OpenText Functional Testing or OpenText Functional Testing for Developers is closed. Additionally, concurrent licenses are released when OpenText Functional Testing or OpenText Functional Testing for Developers is idle, with no mouse or keyboard activity, for a specified amount of time.

If you need to use your FT tool without access to the internet, use one of the following instead:

License type	Description
Commuter license	If you know that you will need access to OpenText Functional Testing or OpenText Functional Testing for Developers without internet, check out a **commuter license** beforehand. Commuter license keys are entered once, and provide use for a single installation of OpenText Functional Testing or OpenText Functional Testing for Developers, for a limited time period. The license key is based on the machine identification, and is specific for the computer making the request.

License type	Description
Remote commuter license	If you find yourself without access to the internet unexpectedly, have another user with a connection check out a commuter license for you. This is called a **remote commuter license**, and must be sent to you for you to use with your FT tool.

Both commuter and remote commuter licenses expire at 23:59 of the expiration day. After the commuter license expires, OpenText Functional Testing and OpenText Functional Testing for Developers automatically return to the previously used license type.

> **Tip:** You can track license usage across your network (for your FT tools as well as other products). For details, see the AutoPass License Server online documentation.

See also:

- AutoPass License Server online documentation

License editions

ADM Functional Testing tools support a variety of license editions, each bundled with a different subset of functional testing features.

In this section:

- Supported license editions 79
- Licensing fallback mechanism 80

Supported license editions

The following table describes the products available with each license edition.

	License names	License names	License names
Includes the use of these products:	**UFT Ultimate***	**UFT One**	**UFT Developer**
OpenText Functional Testing	✓	✓	✗
OpenText Functional Testing for Developers	✓	✓	✓
OpenText Sprinter	✓	✓*	✗
BPT	✓*	✓*	✗
OpenText Functional Testing Lab (for functional testing purposes only)	✓	✗	✗

Additionally, use a **UFT Runtime Engine** license when you need to run OpenText Functional Testing or OpenText Functional Testing for Developers tests only.

The **UFT Runtime Engine** license does not enable you to create or edit tests, or access the OpenText Functional Testing IDE or OpenText Functional Testing for Developers IDE plug-ins.

*** Note:**

- The **UFT Ultimate** license is no longer available for purchase and is supported for existing customers only.
- The **UFT Ultimate** license is available only as a concurrent license.
- OpenText Sprinter is available for **UFT Ultimate** or **UFT One** concurrent licenses only.
- When using BPT with OpenText Functional Testing, you must also have a BPT license on your ALM server.

Licensing fallback mechanism

When starting OpenText Functional Testing or OpenText Functional Testing for Developers, the AutoPass License Server attempts to consume the exact license edition configured on the machine, such as the **UFT One** or **UFT Developer**.

If you are concerned about the availability of the license edition configured on your tool's machine, modify this configuration as described in "Configure the licensing fallback mechanism" on page 93.

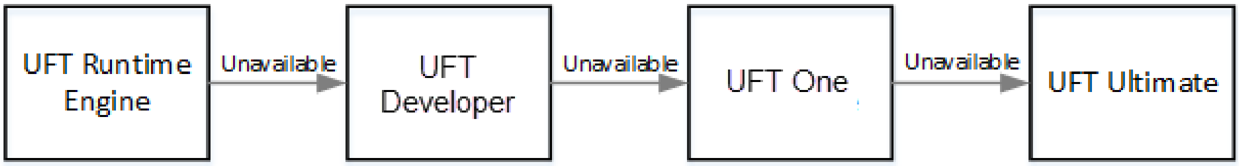
When the fallback mechanism is enabled, licenses are consumed as follows:

When starting OpenText Functional Testing

- If you have a **UFT One** license installed, the License Server looks for the **UFT Ultimate** license as a fallback.
- If you have a **UFT Runtime Engine** or **UFT Developer** license installed, no fallback is supported.

When starting OpenText Functional Testing for Developers

When starting the OpenText Functional Testing for Developers runtime engine, licenses are consumed in the following order on the License Server, starting with the license configured on your machine:



Note:

- The licensing fallback mechanism is relevant only when working with concurrent licenses.
- The licensing fallback mechanism is disabled by default.

Sample scenario 1: UFT Developer license configured on the OpenText Functional Testing for Developers machine

If the **UFT Developer** license is configured on the machine, but there is no available **UFT Developer** license on the License Server, OpenText Functional Testing for Developers tries to consume a **UFT One** license.

In turn, if no **UFT One** license is available, OpenText Functional Testing for Developers tries to consume a **UFT Ultimate** license.

Sample scenario 2: UFT Runtime license configured on the OpenText Functional Testing for Developers machine

If the **UFT Runtime Engine** license is configured on the OpenText Functional Testing for Developers machine but there is no available **UFT Runtime Engine** license, OpenText Functional Testing for Developers tries to consume a **UFT Developer** license.

In turn, if there is no available **UFT Developer** license, OpenText Functional Testing for Developers tries to consume a **UFT One** license.

See also:

- "Licensing" on page 75
- "Configure license behavior" on page 92
- "Licensing FAQs" on page 96

Manage licenses with the wizard

The Functional Testing License Wizard enables you manage licenses for OpenText Functional Testing or OpenText Functional Testing for Developers.

Installing licenses requires administrator permissions.

In this section:

Set seat license mode

Set seat license mode when you have a machine-specific license, based on a specific locking code per computer. For more details, see "Seat vs. concurrent licenses" on page 77.

1. Access the wizard from the **Start** menu or your <**OpenText Functional Testing / OpenText Functional Testing for Developers installdir>\bin\HP.UFT.LicenseInstallationWizard.exe**.
2. In the License Wizard start screen, select **Seat license**.

3. In the Seat License installation screen, do one of the following:
   - Click **Load License Key File** and select your license key **.dat** file.

     Paste the license key in the edit field.
   - If you don't yet have a license key, follow the instructions in the expanded **How can I get a license key file** section.
4. Verify that the license key is valid, and click **Install**.
5. When complete, restart OpenText Functional Testing or OpenText Functional Testing for Developers to apply the new license.

Set concurrent license mode (wizard)

Set the concurrent license mode to have OpenText Functional Testing consume a concurrent license from the AutoPass License Server.

For more details, see "Seat vs. concurrent licenses" on page 77.

Prerequisites

- You must have an OpenText Functional Testing license installed on an AutoPass License Server.

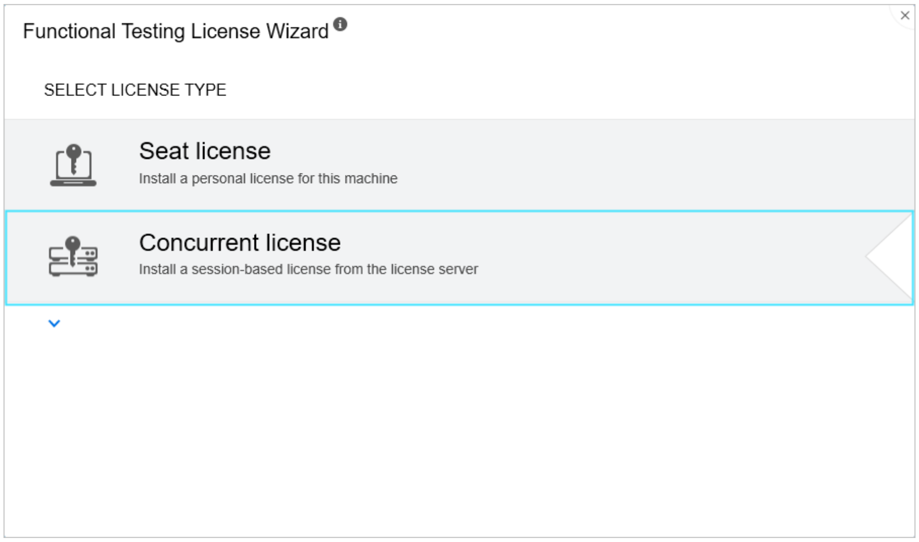
  For details, see the AutoPass License Server online documentation.

  > **Note:** If you need to install OpenText Functional Testing and the License Server, you must install OpenText Functional Testing from the Full UFT One DVD Release or the Core UFT One DVD Bundle.

- Make sure you are connected to the network and can access the AutoPass License Server.

Set the concurrent license mode

1. Access the wizard from the **Start** menu or your <**OpenText Functional Testing / OpenText Functional Testing for Developers installdir>\bin\HP.UFT.LicenseInstallationWizard.exe**.
2. In the License Wizard start screen, select **Concurrent license**.



Functional Testing License Wizard

SELECT LICENSE TYPE

Seat license
Install a personal license for this machine

Concurrent license
Install a session-based license from the license server

3. In the Concurrent License Installation screen:

   Enter the **License Server address** and the **port number**.

   Default port = **5814**

   > **Note:** The address format must be identical to the one used in the **Main** tab of the License Server Configuration pane. For details, see the AutoPass License Server online documentation.

   To use a proxy server for this connection, click **Proxy Settings** and enter the proxy connection details.

   You can use the **System proxy** or an **HTTP proxy**, and specify credentials for proxy authentication, if needed.

4. Provide tenant information if you enter an address of a multi-tenant license server.

   Click **Provide tenant info for a multi-tenant license server**, and enter the tenant ID and authentication token.

   For information on how to assign an authentication token to a tenant in AutoPass License Server, see AutoPass License Server online documentation.

5. Click **Connect** to connect to the License Server.

6. From the product license drop-down list, select the appropriate license and click **Install**.

7. If OpenText Functional Testing or OpenText Functional Testing for Developers was running while you defined the license consumption, restart it to apply the new license.

Check out and consume a Commuter license

Check out and consume a commuter license

Commuter licenses can be checked out only if your License Server has available concurrent licenses.

1. **Prerequisite**: Make sure you are connected to the network and can access the AutoPass License Server.

   Alternatively, if you cannot access the License Server, see "Check out and consume a Remote Commuter license" on page 87.

2. Access the wizard from the **Start** menu or your <**OpenText Functional Testing / OpenText Functional Testing for Developers installdir>\bin\HP.UFT.LicenseInstallationWizard.exe**.

3. In the License Wizard start screen, select **Additional Options > Commuter License**.

4. In the Commuter License Installation screen, enter the License Server address in the following format:

   <license server address>:<port>

   Default port = **5814**

   > **Note:** The address format must be identical to the one used in the **Main** tab of the License Server Configuration pane.
   >
   > For details, see the AutoPass License Server online documentation.

   To use a proxy server for this connection, click **Proxy Settings** and enter the proxy connection details.

   You can use the **System proxy** or an **HTTP proxy**, and specify credentials for proxy authentication, if needed.

5. Click **Connect** to connect to the License Server.
6. After the list of available licenses is displayed, ensure that **Available** is selected below the License Server address field.
7. From the list of available licenses, select the licenses you need.
8. In the **Check out licenses for (days)** field, enter the number of days for which you need the commuter license.

   Maximum = 365 days
9. Click **Check Out**, and then **Next** to define the license consumption.
10. If OpenText Functional Testing or OpenText Functional Testing for Developers was running while you defined the license consumption, restart it to apply the new license.

Return a Commuter license

When you are done with a license, return it back in to the license server.

This process checks in all of the licenses that are checked out. If you still need some of these licenses, check them out again.

1. **Prerequisite**: Make sure you are connected to the network and can access the License Server.

   Alternatively, if you cannot access the License Server, see "Check out and consume a Remote Commuter license" on the next page.
2. Access the wizard from the **Start** menu or your <**OpenText Functional Testing / OpenText Functional Testing for Developers installdir>\bin\HP.UFT.LicenseInstallationWizard.exe**.
3. In the License Wizard start screen, select **Additional Options** > **Commuter License**.
4. In the Commuter License Installation screen, the License Server address should already be displayed and connected.

   If needed, enter the License Server address in the following format:

   <license server address>:<port>

   Default port = **5814**

> **Note:** The address format must be identical to the one used in the **Main** tab of the License Server Configuration pane.
>
> For details, see the AutoPass License Server online documentation.

5. In the area that lists the licenses, ensure that **Checked Out** is selected.

   For example:

6. Click **Check In All Licenses**, and then **Next**. The list of checked out licenses is cleared.

Check out and consume a Remote Commuter license

Check out and consume a remote commuter license

Remote commuter licenses can be checked out only if your License Server has available concurrent licenses.

1. Access the wizard from the **Start** menu or your <**OpenText Functional Testing / OpenText Functional Testing for Developers installdir>\bin\HP.UFT.LicenseInstallationWizard.exe**.
2. In the License Wizard start screen, select **Additional Options > Remote Commuter license**.
3. In the Remote Commuter License Installation screen, ensure that **Generate Request File** is selected.
4. From the list of available licenses, select the license you need.
5. In the **Check out licenses for (days)** field, enter the number of days for which you need the commuter license.

   Maximum = 365 days
6. Click **Generate Request File**.
7. Click the link that appears below this button to open the folder containing the request file.

Send the generated **.lcor** request file to a License Server administrator or to a user with access permissions to the License Server.

The other user must access the Licensing Server to check out and send you a license key file.

8. When you receive the license key file, save it locally.

   Click **Install License**, and click **Choose File** to browse to the text file you received.

9. Click **Install** to install the license.
10. If OpenText Functional Testing or OpenText Functional Testing for Developers was running while you defined the license consumption, restart it to apply the new license.

Return a Remote Commuter license

Perform this procedure after a License Server administrator has checked out your license.

1. Access the wizard from the **Start** menu or your <**OpenText Functional Testing / OpenText Functional Testing for Developers installdir>\bin\HP.UFT.LicenseInstallationWizard.exe**.
2. In the License Wizard start screen, select **Additional Options** > **Remote Commuter license**.
3. In the Remote Commuter License Installation screen, ensure that **Generate Request File** is selected.
4. In the Generation screen, click **Generate and Save Check In Request**, and save the **.lcir** check in request file.
5. Click **Next** to uninstall the license.

The license wizard reports that the remote commuter license is uninstalled. OpenText Functional Testing or OpenText Functional Testing for Developers reverts to the previous license type as the active license.

See also:

- "Seat vs. concurrent licenses" on page 77

Manage licenses with the command line

Consume and verify the statuses of seat or concurrent licenses directly from the command line. Installing licenses requires administrator permissions.

In this section:

- Run the License Installer from the command line 89
- Define seat licenses with the command line 89
- Consume concurrent licenses with the command line 90

Run the License Installer from the command line

Run the License Installer, **LicenseInstall.exe**, as follows:

```
"<OpenText Functional Testing or OpenText Functional Testing
for Developers installation
directory>\bin\HP.UFT.LicenseInstall.exe"
```

Append the relevant command and set of parameters described below:

- "Define seat licenses with the command line" below
- "Consume concurrent licenses with the command line" on the next page

Define seat licenses with the command line

Define seat licenses in the command line by running the License Installer and appending the following:

```
seat "<license key string>"
```

For example:

```
"C:\Program Files (x86)\OpenText\Functional
Testing\bin\HP.UFT.LicenseInstall.exe" seat "<key> \" OpenText UFT
One"
```

Note:

- If the license key contains a quotation mark character (**"**) in the license key string, add a backslash character (****) before the quotation mark.
- If the license key file is saved locally, run the License Installer, appending the following code and wrapping the path to the license key file in quotes:

  ```
  seat "<path to the license key file>"
  ```

  For example:

  ```
  "C:\Program Files (x86)\OpenText\Functional
  Testing\bin\HP.UFT.LicenseInstall.exe" seat "Downloads\UFT-
  licfile.dat"
  ```

For more details, see "Seat licenses" on page 77.

Consume concurrent licenses with the command line

These steps configure OpenText Functional Testing to consume concurrent licenses installed on the AutoPass License Server.

Verify available licenses on the AutoPass License Server

Run the License Installer, appending the following command:

```
licenses <server name/address>:<port> [/tenantid:"xx"
/tenanttoken:"xx"]
```

For example:

```
"C:\Program Files (x86)\OpenText\Functional
Testing\bin\HP.UFT.LicenseInstall.exe" licenses 11.11.111.111:5814
/tenantid:"tenant1" /tenanttoken:"6wxTn89A6BSr04c+nPYstA=="
```

Note: tenantid and **tenanttoken** are required only when you verify available licenses on a multi-tenant License Server.

The available licenses are displayed by unique ID and version.

Consume a concurrent license

1. Run the License Installer to verify which licenses are available on the AutoPass License Server, as described above.

   The available licenses are displayed by unique ID and version.

2. Run the License Installer again, this time appending the following command and parameters:

   ```
   concurrent <license ID> <license version> <server
   address>:<port> [/tenantid:"xx" /tenanttoken:"xx"] [/force]
   ```

   For example:

   ```
   "C:\Program Files (x86)\OpenText\Functional
   Testing\bin\HP.UFT.LicenseInstall.exe" concurrent 10594 1
   11.11.111.111:5814 /tenantid:"tenant1"
   /tenanttoken:"6wxTn89A6BSr04c+nPYstA=="
   ```

Parameters	Description
address	The **address** format must be identical to the one used in the **Main** tab of the AutoPass License Server Configuration pane. For details, see the AutoPass License Server online documentation.
port	Optional**.** The Default port for the server is **5814**.
/tenantid	The ID of an AutoPass License Server tenant. This is required only when your License Server has enabled multi-tenancy.
/tenanttoken	The token assigned to the tenant. This is required only when your License Server has enabled multi-tenancy.

Parameters	Description
/force	Optional. **/force** saves the license installation information even if the current installation fails. In subsequent sessions, OpenText Functional Testing or OpenText Functional Testing for Developers will check the listed license server for the listed license.

Modify server connection protocol

Run the License Installer, appending:

```
config protocol.primary <protocol>
```

where **<protocol>** is **http** or **https** as needed.

See also:

- "Seat vs. concurrent licenses" on page 77

Configure license behavior

This section describes how to configure your OpenText Functional Testing license behavior.

If you are installing OpenText Functional Testing for Developers on Linux or Mac, or OpenText Functional Testing for Developers standalone, see the OpenText Functional Testing for Developers Help Center instead.

> **Note:** Some folder paths intentionally include previous company branding, due to backward compatibility considerations.

In this section:

- General license settings 93
- Configure the licensing fallback mechanism 93
- Configure licensing timeouts 95

General license settings

General licensing behavior is managed in the AutoPass license configuration file, located on your OpenText Functional Testing or OpenText Functional Testing for Developers machine. This file includes details about supported options and values.

The file is located at: **C:\ProgramData\OpenText\UFT\License\autopass.txt**

> **Caution:** Configure this file with caution. Incorrect configuration may prevent the products consuming the licenses from starting, or cause them to behave unexpectedly.

Additional configurations include:

- "Configure the licensing fallback mechanism" below. Perform this procedure if your concurrent license server has multiple license editions installed, and you want to always ensure that your product can find an available license.
- "Configure licensing timeouts" on page 95. Define the timeout period after which your license is released.

Configure the licensing fallback mechanism

Define whether your system uses the licensing fallback mechanism for OpenText Functional Testing and OpenText Functional Testing for Developers as follows:

1. On your AutoPass License Server machine, browse to the **C:\ProgramData\autopass\apls\licenseserver\data\conf\UFT.xml** file.

   > **Note:** This file is available with AutoPass versions 9.3 or higher.

2. Edit and add keys and values to set the following values to **true**, as needed:

Product	License type	Key
OpenText Functional Testing	Any	**license.fallback.uft.rte**
Runtime Engine	Any	**license.fallback.rte.rte**
OpenText Functional Testing for Developers	UFT Developer	**license.fallback.leanft.leanft**
OpenText Functional Testing for Developers	Runtime Engine	**license.fallback.leanft.rte**

Make sure that you add key entries within the <properties> element and after the <comment> element.

Edit and add keys and values in the following format:

```
<entry key="{Key}">{Value}</entry>
```

> **Example:** To enable the fallback mechanism when you want to use **OpenText Functional Testing** and you have **any** of the license types configured, set the relevant key value to **true**, as follows:
>
> ```
> <entry key="license.fallback.uft.rte">true</entry>
> ```

Finding a Runtime Engine license

If the fallback mechanism is enabled, and an available **Runtime Engine** license is found, you will only be able to run your tests, with no creation or editing abilities.

Ensure that you can always access the OpenText Functional Testing IDE or the OpenText Functional Testing for Developers IDE plug-ins by doing one of the following:

- Deactivate the fallback mechanism by setting the key values to **false** (this is the default).
- Contact your License Server administrator to ensure that any OpenText Functional Testing Runtime Engine licenses are blocked or are in use.

For more details, see "Licensing fallback mechanism" on page 80.

Configure licensing timeouts

Define the number of minutes, with no keyboard or mouse input, after which OpenText Functional Testing or OpenText Functional Testing for Developers releases the currently used concurrent license.

Configure a timeout for OpenText Functional Testing or OpenText Functional Testing for Developers

1. On your OpenText Functional Testing or OpenText Functional Testing for Developers machine, open the **LicenseSettings.xml** file for editing.

   The file is located in the **C:\ProgramData\OpenText\UFT\License** folder.

2. Update the following parameters with the number of minutes you want to define for your timeout:

Parameters	Description
LicenseAutoReleaseInterval	The number of minutes after which a confirmation message appears, warning the user that the license is about to time-out.
ConfirmLicenseReleaseTimeout	The number of minutes after which the confirmation message closes, and the license is released.

Configure a timeout for your AutoPass concurrent license server

On your AutoPass License Server machine, browse to the **UFT.xml** file at **C:\ProgramData\autopass\apls\licenseserver\data\conf\UFT.xml**.

Open the file for editing, and add the following line of code:

```
<entry key="autorelease.interval"><#></entry>
```

where **#** is the number of minutes of inactivity.

> **Example:** The following syntax defines that your license is released after 10 minutes of inactivity.

```
<entry key="autorelease.interval">10</entry>
```

See also:

- "Licensing FAQs" below
- "Known issues with licensing" on page 100
- "Licensing" on page 75

Licensing FAQs

This section answers a number of frequently asked questions about using and installing Functional Testing licenses.

> **Note:** Some folder paths intentionally include previous company branding, due to backward compatibility considerations.

In this section:

- OpenText Functional Testing Help Center licensing scope 96
- Which license should I install? 97
- How do I install the AutoPass License Server? 97
- If I am using concurrent licenses, how do I connect to the License Server? 97
- How do I install licenses if I am deploying OpenText Functional Testing across an enterprise network? 98
- How do I manage the concurrent licenses on the License Server? 98
- Can I configure license behavior myself? 98
- Can I use the AutoPass License Server with a proxy? 99
- What is a cleanup license? 99
- My demo license is expiring early. What can I do? 99

OpenText Functional Testing Help Center licensing scope

This guide describes how to access licenses on the AutoPass License Server from OpenText Functional Testing and OpenText Functional Testing for Developers.

For full details on AutoPass License Server capabilities, such as proxy settings, license installation and management, and user management, see the AutoPass License Server online documentation.

Which license should I install?

Use the following table to help determine which type of license to install. For details about license types, see "Licensing" on page 75.

Scenario	License Type to Install
Are you assigned a specific license (with its own unique license key)?	Seat
Are you part of a group that uses licenses on an as-needed basis?	Concurrent. You will need the IP address of your License Server where the licenses are installed.
Are you assigned the IP address from which to check out a license?	Concurrent
Are you traveling and will not have access to a license server?	Concurrent Commuter
Are you already traveling and cannot access the License Server to get a license?	Remote Commuter

How do I install the AutoPass License Server?

Download the AutoPass License Server from the Marketplace (login required).

For full details, see the AutoPass License Server online documentation.

If I am using concurrent licenses, how do I connect to the License Server?

Run the Functional Testing License Wizard and enter the License Server IP address. This checks the connection to the License Server, and also provides a list of possible licenses to install.

After installing your concurrent license, OpenText Functional Testing or OpenText Functional Testing for Developers checks the specified License Server address each time OpenText Functional Testing or the OpenText Functional Testing for Developers runtime engine starts, taking the requested license.

For more details, see "Manage licenses with the wizard" on page 82.

How do I install licenses if I am deploying OpenText Functional Testing across an enterprise network?

OpenText Functional Testing provides a command-line tool that enables you to install OpenText Functional Testing licenses without using the License Wizard interface.

For details on the commands to install these licenses, see "Manage licenses with the command line" on page 89.

The command line license installation is supported for seat and concurrent licenses.

How do I manage the concurrent licenses on the License Server?

The AutoPass License Server has a full Web-based interface that enables you to install, manage, administer, and track usage of all your licenses (both concurrent and commuter).

For details, see the AutoPass License Server online documentation.

Can I configure license behavior myself?

Yes. For details, see "Configure license behavior" on page 92.

Can I use the AutoPass License Server with a proxy?

Yes.

If you are using the Functional Testing License Wizard, you can configure the proxy connection in the Wizard. For details, see "Manage licenses with the wizard" on page 82.

Alternatively, you can set the proxy settings in the **autopass.txt** file, located on your OpenText Functional Testing or OpenText Functional Testing for Developers machine. The file is located in the **C:\ProgramData\OpenText\UFT\License** folder.

See the comments inside this file for details on setting the proxy settings. Be sure to uncomment the relevant lines and define their values.

> **Note:** For Linux/Mac installations of OpenText Functional Testing for Developers, see the OpenText Functional Testing for Developers Help Center.

What is a cleanup license?

If your computer is clock-tampered after installing the License Server, both the License Server and OpenText Functional Testing's or OpenText Functional Testing for Developers's connection to the License Server do not work.

In this case, you must get a cleanup license for your License Server. This enables you to reset all license capabilities.

For details on cleanup licenses, contact your OpenText Functional Testing license supplier.

My demo license is expiring early. What can I do?

If you are having problems with the 30-day trial license period, ensure the following:

- Ensure that you have full permissions to the OpenText Functional Testing or OpenText Functional Testing for Developers installation folder and all its subfolders.
- Ensure that you have not changed the system time. If you have moved the system time, the license mechanism can reduce the trial period based on the number of days that were back-dated.

Known issues with licensing

Relevant for: GUI testing and API testing

The following known issues exist when working with Functional Testing licenses:

Known issues	Description
OpenText Functional Testing and OpenText Functional Testing for Developers concurrent installations	If you installed OpenText Functional Testing for Developers from the OpenText Functional Testing setup program, and you are using a seat license for OpenText Functional Testing, OpenText Functional Testing for Developers uses the same license. In such cases, you cannot run both OpenText Functional Testing and OpenText Functional Testing for Developers at the same time.
Modifying the computer date	If you install a time-limited seat license, do not modify the date on your computer. Doing so will block your active seat license and prevent future seat license installations on that computer. For questions about this issue, contact your OpenText Functional Testing license supplier.
NAT	The License Server does not support the use of Network Address Translation (NAT).
Demo licenses	Demo licenses are not included in concurrent licenses, which require an active connection to the AutoPass License Server and a license key installed.
Changing types	You must have administrator permissions to change the license type from seat to concurrent or vice versa.

If you did not find a solution to your issue here, search for license issues in the Community.

See also:

- "Licensing" on page 75
- "License editions" on page 79
- "Manage licenses with the wizard" on page 82
- "Manage licenses with the command line" on page 89
- "Licensing FAQs" on page 96

Security Reference

This section is designed to help users who deploy and manage OpenText Functional Testing instances in a secure manner in the modern enterprise. The objective is to help you make well-informed decisions about the various capabilities and features that OpenText Functional Testing provides to meet modern enterprise security needs.

Security requirements for the enterprise are constantly evolving and this information should be viewed as OpenText's best effort to meet those stringent requirements. If there are additional security requirements that are not covered in the help, please open a support case with the OpenText support team to document them.

The following sections discuss potential security issues when using OpenText Functional Testing:

- "Installing and using your product securely" below
- "Installation and deployment security" on page 105
- "DCOM configuration settings" on page 105
- " Connecting to OpenText Application Quality Management" on page 108
- "Securing test information" on page 109
- "Working with Safari on Mac" on page 110
- "Connecting to OpenText Functional Testing Lab" on page 111

Installing and using your product securely

OpenText Functional Testing is a desktop application installed on a single computer or on multiple computers in a business network. Security issues are similar to those of other Windows-based applications.

OpenText Functional Testing can potentially be used to record user actions and/or network communications. Therefore, we strongly recommend running it on dedicated test machines that do not contain or provide access to sensitive information. Additionally, you should thoroughly review your lab network topology and access permissions before using this program.

You must have specific permissions when installing and running OpenText Functional Testing. For a list of these permissions, see "Required access permissions" on page 23.

For a list of ports that OpenText Functional Testing uses for communication, see the following knowledge base article: https://portal.microfocus.com/s/article/KM000003271.

When installing, the following security settings are provided:

- You can install and run OpenText Functional Testing with the computer's User Account Control (UAC) enabled.
- During installation, you can make the following allowances:
  - Specify whether to configure DCOM settings that enable remote computers to access OpenText Functional Testing to run a test from OpenText Application Quality Management or using automation.
  - Specify whether to change browser policies to allow installing the browser extension that enables testing web applications.

  You can also adjust these settings after installation.
- You can securely store important and sensitive information about the applications you are testing.

See also:

- "DCOM configuration settings" on the next page
- "Browser policy settings" on page 107

Installation and deployment security

OpenText Functional Testing can be installed with UAC enabled. This includes the installation of all prerequisite software, as well as all OpenText Functional Testing add-ins, the OpenText Functional Testing Add-in for ALM, and installation configurations.

When running the installation, note the following:

- If you install the OpenText Functional Testing Add-in for ALM as part of the installation, and your computer has UAC enabled, you must run an additional installation for the Add-in for ALM following the OpenText Functional Testing installation.
- The following configuration options are enabled by default:
  - The option to configure DCOM settings for OpenText Application Quality Management integration with OpenText Functional Testing.
  - The option to configure your browser policies to allow installing OpenText Functional Testing's browser extension on your browsers.

  If you want to clear these options, you can do so in the installation wizard.

For full details on secure installation and deployment, see the "Enterprise deployment" on page 11.

See also:

- "DCOM configuration settings" below
- "Browser policy settings" on page 107

DCOM configuration settings

You can configure DCOM settings to enable outside computers or OpenText Application Quality Management to work with and run tests on the OpenText Functional Testing computer. This configuration can be performed during the installation or after the installation.

For details, see "OpenText Functional Testing Configuration screen" on page 30.

There are two possible options for configuring DCOM settings:

Option	Description
Enable running OpenText Functional Testing remotely from OpenText Application Quality Management	This sets the DCOM configuration to enable an ALM project to access your computer and run tests on the computer. **Note**: There are additional settings you must configure in your ALM project to determine the access level to the OpenText Functional Testing computer. For details, see " Connecting to OpenText Application Quality Management" on page 108.
Enable running OpenText Functional Testing remotely from Automation Scripts	This sets the DCOM configuration to enable any computer to run tests using the OpenText Functional Testing Automation Object Model. Enabling this configuration can present a security risk as it allows the remote computer full access to the OpenText Functional Testing computer.

> **Note:** You must perform these configurations to run tests from OpenText Application Quality Management or with automation, so care must be taken to determine the need for these settings.

When configuring DCOM settings, the following settings are recommended to ensure security for the computer running OpenText Functional Testing:

- Remove DCOM permissions for broad groups in the DCOM settings, such as the **Anonymous Login**, **Everyone**, **Interactive**, and **Network** groups.
- Give permissions only to specific groups or users.

See also:

- "Installing and using your product securely" on page 103
- "Installation and deployment security" on the previous page

Browser policy settings

To enable testing web applications, the OpenText Functional Testing Agent browser extension must be installed on your browsers.

UFT Developer 23.4, 24.2, and 24.4: The extension is named OpenText UFT Agent.

Versions 2023 and earlier: The extension is named Micro Focus UFT Agent.

Your browser policies must allow this browser extension installation. You only need to allow the installation of this one specific extension on each browser you want to test. Browser policies can be configured by the installation program, or you can adjust them manually yourself. For details, see the "OpenText Functional Testing Configuration screen" on page 30.

> **Caution:** Make sure that your company's security policies permit you to make this change.

The installation option that controls the browser policy update:

Set Chrome, Chromium Edge, Firefox options

This option is available on versions 2021 and later.

This option updates your browser policies and adds the OpenText Functional Testing Agent extension to your browsers' allowlist or installation list.

This enables automatic installation the extension in Chrome, Edge, and Firefox.

For a list of the extensions added, see the known issues topic for Chrome, Edge and Firefox in the relevant version of the Help Center.

See also:

- "Installing and using your product securely" on page 103
- "Installation and deployment security" on page 105

Connecting to OpenText Application Quality Management

When connecting, OpenText Functional Testing connects with a "super user" permission level, regardless of the specific user permissions assigned to you in OpenText Application Quality Management. This enables you to use all aspects of the application regardless of the privileges allotted to you for working in an ALM project.

> **Note:** This section is only relevant if you have enabled communication between OpenText Functional Testing and ALM project by setting the relevant DCOM settings. For details, see "DCOM configuration settings" on page 105.

Default access level

The ALM project uses the parameter *ALLOW_LEGACY_INTEGRATION_MODE*. By default, this parameter is deactivated, and your activities in OpenText Application Quality ManagementALM by OpenText Functional Testing are limited by the assigned user permissions in your ALM project.

For full details on the parameter *ALLOW_LEGACY_INTEGRATION_MODE*, see the OpenText Application Quality Management Help Center.

In addition, if you are running GUI tests from the ALM Test Lab, you must select the **Allow connections from computers running any version of Remote Desktop (less secure)** option in the Windows Remote Settings (**Control Panel > System > Remote Settings**) . Failure to enable this option will result in the test run stopping when the Remote Desktop session is disconnected.

Connecting to OpenText Application Quality Management using Single Sign-On (SSO)

To connect to an ALM server that requires Single Sign-On (SSO), you provide the SSO credentials.

When you run tests manually from OpenText Functional Testing, you enter the credentials directly in an SSO login page provided by OpenText Application Quality Management. OpenText Functional Testing never stores these credentials.

When you run OpenText Functional Testing using an automation script and connect to OpenText Application Quality Management, you must enter the credentials in the script. The credentials you enter are stored in a text file. We therefore strongly recommend that you always encode this information before entering it in the script. For details, see the *Automation Object Model Reference*.

Securing test information

Sometimes, a test must contain sensitive information, such as user names or passwords to access the application being tested. OpenText Functional Testing enables you to make this data harder to access.

GUI testing

- Use the **SetSecure** test object method instead of the normal **Set** method to enter passwords into password fields.

  OpenText Functional Testing automatically records a **SetSecure** step when you record on a standard password field.

- When retrieving password data from another source during a run, store the data in a variable and then use the **Crypt.Encrypt** method to encrypt the value before using it with a **SetSecure** step.
- When using the Data Table to provide data for a password field, use the **Data > Encrypt** right-click option to encrypt the data.
- To generate an encoded value for a password field, use the Password Encoder tool.

> **Note:** The above tools and methods do not use a global standard for encryption. The encryption is not considered nor intended to be fully secure.

> Its purpose is only to ensure that passwords will not be readily visible on the screen while recording, editing or running a test or component.
>
> If you are using real customer data or other sensitive information, you should take additional steps to ensure the security of that data.

API testing

- Use event handlers to encrypt passwords in your API tests.
- Set security properties for accessing your Web service calls during API tests which contain HTTP or Web Service call activities.

See also:

- OpenText Functional Testing Help Center

Working with Safari on Mac

When OpenText Functional Testing connects to a remote Mac computer, it can access the Safari application and perform steps on Web applications running in Safari. Therefore, it is important to secure this connection, to prevent inappropriate access to your Mac and Web pages that the Mac can access.

When OpenText Functional Testing communicates with the Mac, OpenText Functional Testing acts as a client and the OpenText Functional Testing Connection Agent acts as a server.

You can secure this communication on different levels:

- Set up client authentication by defining a passphrase for OpenText Functional Testing to use when contacting the Mac.
- Secure the communication between OpenText Functional Testing and the OpenText Functional Testing Connection Agent by requiring that they use an SSL connection, and providing the necessary certificate and key files for SSL communication.

See also:

- OpenText Functional Testing Help Center

Connecting to OpenText Functional Testing Lab

Connect to OpenText Functional Testing Lab to record and run tests on mobile devices.

It is important to secure this connection to prevent inappropriate access to the OpenText Functional Testing Lab server and any connected devices or browsers.

To use a secure connection when connecting to OpenText Functional Testing Lab:

1. Install the SSL certificates on your OpenText Functional Testing machine.
2. In the OpenText Functional Testing Options dialog box:

   In the pane for setting up the connection to the device and browser lab, (**Tools > Options > GUI Testing > Functional Testing Lab**, **Digital Lab**, or **Mobile**), paste in the secure URL (HTTPS). The default port for secure servers is 8443.
3. Provide the authentication information—either basic username and password authentication or an API access key provided by the OpenText Functional Testing Lab.

See also:

- OpenText Functional Testing Help Center
- OpenText Functional Testing Lab Help Center

Connecting to a Remote AI Object-Detection Service

To maximize the performance of OpenText Functional Testing's AI Object Detection, you can install the AI Object-Detection Service on one powerful computer and connect your OpenText Functional Testing computer to this service remotely.

OpenText Functional Testing connects to the remote AI object-detection service using the https protocol.

Set up OpenText Functional Testing's connection to the remote AI object-detection service in OpenText Functional Testing, in **Tools** > **Options** > **GUI Testing** > **AI Object Detection** > **Service** > **Use remote service**

By default, the remote AI object-detection service uses a self-signed certificate. To secure OpenText Functional Testing's connection with the remote AI object-detection service, use a certificate signed by a recognized CA.

For more details, see the section "Secure the communication using a CA certificate" in the Help Center.

See also:

- Help Center: AI Options
- Help Center: Remote AI object-detection service

Connecting to Cloud OCR vendors

OpenText Functional Testing connects to Cloud OCR vendors such as Google and Baidu to use text recognition services when running tests.

Set up a Baidu or Google Cloud OCR, and obtain an API to use for access. Configure OpenText Functional Testing's connection to the cloud OCR in the OpenText Functional Testing options dialog box or using an Automation Object Model (AOM) script.

- In the options dialog, the API key that you enter is not displayed and you cannot retrieve it from the dialog box.
- In an automation script, you must encode the API key before providing it in the script. To encode the API key, use the Password Encoder tool delivered with OpenText Functional Testing (**<Installdir>\bin\CryptonApp.exe**).

> **Note:** The Password Encoder encodes the API key so that it is not visible in the script but does not encrypt it using a global encryption standard. The encoding is not considered or intended to fully secure the information.
>
> The Password Encoder provides one-way encoding. It generates an encoded string from the API key you provide but cannot be used to retrieve the API key from the encoded result.

OpenText Functional Testing stores only the encoded string and does not display the actual API key anywhere.

See also:

- OpenText Functional Testing Help Center

CI server integration

You can run OpenText Functional Testing tests using CI builds on servers such as Jenkins, Bamboo, and Azure DevOps (formerly known as TFS).

Although any sensitive information is encrypted and encoded, OpenText encourages you to routinely remove unnecessary files generated by your CI builds, which are not removed by the CI plugin.

By not implementing the file removal you may expose your system to increased security risks. You understand and agree to assume all associated risks and hold OpenText harmless for the same.

It remains at all times the Customer’s sole responsibility to assess its own regulatory and business requirements. OpenText does not represent or warrant that its products comply with any specific legal or regulatory standards applicable to Customer in conducting Customer's business.

Routinely remove all files from the following folders:

Server type	File location
Jenkins	**<Jenkins_installdir>\workspace\<job name>**

Server type	File location
Bamboo	**<Bamboo home directory>\xml-data\build-dir\<job name>**
Azure DevOps	**UFTWorking\props**

For information on setting up the integration with a CI server, see the **Integrations** section of the Help Center.

Download Help Center

The OpenText Functional Testing Help Center provides you with comprehensive user assistance.

We recommend that you use the online version of the Help Center for the most recent updates available.

If your organization has restrictions that prevent you from using the online Help Center, switch to the local mode, and save the Help Center files locally on your OpenText Functional Testing machine.

Access your local help by default

To download and configure OpenText Functional Testing to access the local help by default:

1. Download and extract the OpenText Functional Testing Help Center.ZIP file to the **<Installdir>\help** folder on your OpenText Functional Testing machine.
2. Open the **Tools > Options** dialog box, select the **Help** pane, and select **Access local Help**. In this way, when you open the Help from the product, it opens the local help.

API and schema references

Much of the product documentation is not version dependent and is extracted to the **<Installdir>\help\References** folder.

These references are not linked from the product Help menu, and need to be opened manually. To access these references in the extracted help, open the **htm** file (usually **Default.htm**) in the root of the relevant subfolder.

Reference help name	Access the help in this subfolder
OpenText Functional Testing Automation Object Model Reference	**AutomationObjectModel**

OpenText Functional Testing Object Repository Schema Reference	**ObjRepSchema**
OpenText Functional Testing Object Repository Reference	**ObjRepUtil**
OpenText Functional Testing Run Results Schema Reference	**XMLReport**
Extensibility Add-in References: Delphi, Extensibility Accelerator, Java, .NET, Testing Extensibility, Web, and WPF/Silverlight	**Extensibility\<technology_Ext>**