

VMware User Environment Manager Application Profiler Administration Guide

VMware Dynamic Environment Manager 9.7
VMware User Environment Manager



vmware®

You can find the most up-to-date technical documentation on the VMware website at:

<https://docs.vmware.com/>

If you have comments about this documentation, submit your feedback to

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2019 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

Contents

1	Introduction to User Environment Manager Application Profiler	4
2	Installing Application Profiler	5
	Application Profiler System Requirements	5
	Install Application Profiler Manually	6
	Unattended Installation of Application Profiler	6
3	Using Application Profiler	7
	Start a Profiling Session	7
	Profile an Application	8
	Filtering and Optimizing the Analysis Details	9
	Filtering	9
	Optimizing the Flex Configuration File	9
	Editing the Flex Configuration File	10
	Saving the Flex Configuration File	10
	Saving a Flex Configuration File With Predefined Settings	10
	Editing a Profile Archive	11
	Exclusions	11
	Application Profiler Project Files	11
	Advanced Configuration of Application Profiler	11
	Using File System Auto Excludes	13
	Command-Line Arguments	13

Introduction to User Environment Manager Application Profiler

1

Application Profiler is a standalone application that simplifies the creation of Flex configuration files and predefined settings for use with VMware User Environment Manager.

Application Profiler analyzes where an application stores its file and registry configuration. The analysis results in an optimized Flex configuration file, which you can edit in the Application Profiler or use directly in the User Environment Manager environment.

With Application Profiler, you can also create application-specific predefined settings, with which you can set the initial configuration state of applications. Save the Flex configuration file with predefined settings to export the current application configuration state.

Application Profiler is licensed as a VMware User Environment Manager component.

Installing Application Profiler

You can install Application Profiler manually, or you can perform an unattended installation.

Note The Application Profiler MSI file has a digital signature, which the Windows Installer infrastructure validates when the installation starts. The installation process includes a certificate revocation check for which the system requires Internet access. If the Internet connectivity is not sufficient, the installation continues, but only after several timeouts. During the process, the installer seems to hang without providing any feedback.

This chapter includes the following topics:

- [Application Profiler System Requirements](#)
- [Install Application Profiler Manually](#)
- [Unattended Installation of Application Profiler](#)

Application Profiler System Requirements

Use Application Profiler in an environment that corresponds with the environment in which User Environment Manager is deployed.

When you use User Environment Manager on a terminal server (RDSH), install Application Profiler on the equivalent desktop OS.

Application Profiler supports the following Windows versions and editions:

- Windows 7 Professional, Enterprise, and Ultimate x86 and x64 SP1
- Windows Server 2008 R2 Standard and Enterprise x64 SP1
- Windows Server 2012 Standard and Datacenter x64
- Windows 8.1 Professional and Enterprise x86 and x64 with Update
- Windows Server 2012 R2 Standard and Datacenter x64 with Update
- Windows 10 Version 1809 (November 2018 Update) Professional and Enterprise x86 and x64
- Windows Server 2016 Standard and Datacenter x64
- Windows Server 2019 Standard and Datacenter x64

No specific hardware other than the system requirements of the installed applications is required for Application Profiler.

Install Application Profiler Manually

You install Application Profiler by launching a setup wizard that guides you through the installation.

Prerequisites

- Verify that you have administrative privileges on the account where you will run the MSI file.
- Download and extract the MSI file package for your operating system.
- Ensure you installed User Environment Manager on a different machine from where you plan to install Application Profiler.

Procedure

- 1 Run the MSI file that corresponds to your OS architecture, and click **Next**.

Option	Description
x86	VMware UEM Application Profiler 9.6 x86.msi
x64	VMware UEM Application Profiler 9.6 x64.msi

- 2 Read and accept the **End User License Agreement** and click **Next**.
- 3 Select the destination folder where you want to install the application and click **Next**.
VMware recommends you install Application Profiler in the default folder.
- 4 Click **Install**, and after the installation is complete, click **Finish**.

Unattended Installation of Application Profiler

The Application Profiler MSI supports unattended installation by using MSI properties to specify installation parameters. To perform an unattended installation, run the `msiexec` utility from the command line with the following property.

Property	Description
INSTALLDIR	The absolute path to the installation directory. Default value is %ProgramFiles%\Immidio\Application Profiler.

The following command is an example of a custom unattended installation command.

```
msiexec.exe /i "VMware UEM Application Profiler 9.6 x86.msi" /qn INSTALLDIR="D:\Apps\VMware UEM Application Profiler" /l* InstallProfiler.log
```

Using Application Profiler

Application Profiler analyzes at which registry and file system locations the settings for a particular application are stored, to create a Flex configuration file for use with User Environment Manager.

From a high-level perspective, the process is as follows:

- 1 Start Application Profiler.
- 2 From within Application Profiler, start the application to analyze.
In the background, Application Profiler monitors the registry and file system actions of the started application.
- 3 Change the necessary settings in the application to make sure that all application settings are saved, and exit the application.
- 4 Application Profiler stops monitoring and outputs the collected information as a Flex configuration file.

This chapter includes the following topics:

- [Start a Profiling Session](#)
- [Profile an Application](#)
- [Filtering and Optimizing the Analysis Details](#)
- [Editing the Flex Configuration File](#)
- [Saving the Flex Configuration File](#)
- [Editing a Profile Archive](#)
- [Exclusions](#)
- [Application Profiler Project Files](#)
- [Advanced Configuration of Application Profiler](#)
- [Command-Line Arguments](#)

Start a Profiling Session

You can start a profiling session by running Application Profiler on a system that corresponds to the environment in which VMware User Environment Manager is used.

Procedure

- 1 Install the application to profile on the profiling system.
- 2 Log in to the profiling system with an administrator account.
- 3 Start Application Profiler.

Profile an Application

When you profile an application, Application Profiler monitors the changes that the application makes to the registry and the file system. It is important that the application saves its configuration during the analysis session.

Only .EXE files are supported for analysis. Some applications install shortcuts in the start menu that refer to an application document instead of to the executable file of the application. You can profile these applications by browsing to the executable and adding any additional arguments after the application path.

Usually, it is sufficient to modify a few of the settings of the profiled application. Many applications save their full configuration whenever a change is made. You might need to change more of the application features and settings, so that the corresponding files are written to disk. For example, you might change settings such as creating a signature in an email client or adding an entry to the custom dictionary in a word processor.

You cannot select applications for which an application template is provided in the User Environment Manager Management Console for analysis in Application Profiler. For such applications, you can create a Flex configuration file in the Management Console with the appropriate template.

To find the registry and file system definitions for a Flex configuration file, the actual configuration settings are not relevant. Only the locations where the settings are kept in the user profile are important. To create predefined settings, the actual configuration settings do matter, because they become part of the predefined settings archive.

Application Profiler monitors the application you selected and all of the child processes started by that application. Monitoring stops when the main application and all child processes are stopped. For some applications, one or more of those child processes continue to run even when you exit the main application. In that case, you must manually stop the analysis by clicking the **Stop Analysis** button in the Analyzing Application dialog. Stopping the analysis only affects the monitoring by Application Profiler and does not stop the child processes.

Procedure

- 1 Start Application Profiler and click **Start Session**.
- 2 Browse to and select the application to be profiled or select an application from the **All Programs** tree.

You can optionally enter command-line arguments after the application path in the Application text box and specify the application start folder in the Start in text box. Put quotes around the path if necessary. If you select a shortcut from the All Programs tree, all settings are read from the shortcut.

3 Click **OK**.

The **Analyzing Application** window appears and the application starts.

4 Make the necessary changes to the application configuration and exit the application.

The main Application Profiler user interface appears, with the analysis results.

5 (Optional) Adjust the generated settings by modifying the Application Profiler options or making manual changes in the editor.

6 Click **Save** and select one of the options and a location for the Flex configuration file.

You successfully profiled an application.

Filtering and Optimizing the Analysis Details

During a profiling session, Application Profiler collects information by monitoring registry and file system access. From this information, a Flex configuration file is generated, using optimizations and filtering options.

The full-fidelity analysis details are available until you start a new profiling session or exit Application Profiler.

Filtering

By default, file system access to locations outside the user profile is filtered out, because User Environment Manager does not support managing files outside the user profile. This behavior can be changed by selecting the Show Unsupported File Access setting check box in Application Profiler.

Access to the HKEY_LOCAL_MACHINE registry hive is excluded, unless you select the Support HKLM setting check box. User Environment Manager only supports managing HKLM registry information when used together with application virtualization.

After this initial filtering, the exclusions are applied. For both the registry and the file system, by default, a number of paths are defined to exclude settings that are typically not application-specific. You can edit these exclusions by clicking **Manage Exclusions** in the Registry or File System sections on the **Program Analysis** tab. Whether the exclusions are in effect is controlled by selecting the **Enable Registry Exclusions** and **Enable File Exclusions** check boxes on the **Settings** tab.

If you select **Use Deepest Registry Path**, only the most specific registry paths are kept.

Optimizing the Flex Configuration File

After the filters are applied, an optimization step can be performed.

In registry terms, optimizing means that the access to separate registry keys and values below a common registry key HKCU*Software*\ *Vendor* is converted to a single reference to the HKCU*Software*\ *Vendor* registry tree. In file system terms, a similar optimization applies to file and folder access below a common <AppData>\ *Vendor* or <LocalAppData>\ *Vendor* folder.

The Optimization Level setting controls how many key or folder levels remain after optimization. By default, both are set to 1, resulting in paths such as HKCU*Software\Vendor* and <AppData>\ *Vendor*. To manage separately multiple applications that the same vendor produces, where configuration is stored in *... \Vendor\Application\...* paths, the level must be increased to 2. With this configuration, each Flex configuration file manages only the settings for a particular application.

Editing the Flex Configuration File

The generated Application Profiler Flex configuration file can be edited, similar to the editor in the User Environment Manager Management Console.

Section headers and folder tokens can be inserted, by clicking the **Section** and **Folder Token** buttons on the **Program Analysis** tab. You can also type a [character in the editor to display a drop-down menu from which you can select a section header. You can also type a < character in the editor to display a drop-down menu from which you can select a folder token.

If you select the **Browse Local Profile** check box on the **Settings** tab, file system and registry paths are auto-completed based on the actual contents of the user profile.

For full information about the Flex configuration files format, see the *VMware User Environment Manager Administration* guide.

Saving the Flex Configuration File

Save the Flex configuration files you create using Application Profiler to a test environment, to verify that all settings are correctly managed before deploying the configuration files on your production systems.

If you configured **Default Save Path** on the **Settings** tab, the **Save As** dialog box always opens to that path. If you selected **Save Icon** on the **Settings** tab, Application Profiler saves the icon of the analyzed application next to the Flex configuration file.

The application you started in the profiling session is included in the Flex configuration file as a DirectFlex executable and DirectFlex is enabled. This behavior does not apply to App-V 4.x applications. For App-V 4.x applications, you must manually add integration in the User Environment Manager Management Console. For details, see the *VMware User Environment Manager Administration* guide.

Saving a Flex Configuration File With Predefined Settings

You can create predefined settings to be saved with your Flex configuration file.

If you select **Save Config File with Predefined Settings**, a profile archive is created, to use for predefined settings when a user logs in. For information about predefined settings, see the *VMware User Environment Manager Administration* guide.

If you select the **Display Predefined Settings Size** check box on the **Settings** tab, the size of the generated profile archive is displayed after you create it.

If you select **Save Predefined Settings Only**, predefined settings are created without saving the Flex configuration file.

Editing a Profile Archive

You can verify which settings are saved as predefined settings or modify them as necessary.

Click the **Edit Profile Archive** button to select a profile archive, containing the predefined settings. After you select the archive, Windows Explorer opens a folder with the extracted settings.

Exclusions

By default, a number of paths are configured that do not contain application-specific configuration settings and therefore must not be included in the contents of a Flex configuration file. You can modify the default settings and add your own exclusions. Exclusions are not case-sensitive and can contain * wildcards.

Registry Exclusions

By clicking **Manage Exclusions** in the Registry section of the **Program Analysis** tab, you can configure which registry paths to filter out. The exclusions are only applied when you select the **Enable Registry Exclusions** check box on the **Settings** tab.

File System Exclusions

By clicking **Manage Exclusions** in the File System section of the **Program Analysis** tab, you can configure which file system paths to filter out. The exclusions are only applied when you select the **Enable File Exclusions** check box on the **Settings** tab.

Application Profiler Project Files

After Application Profiler has analyzed an application, the analysis results are filtered and optimized and a Flex configuration file is generated. The full-fidelity analysis details are available until you start a new profiling session or exit Application Profiler.

The full analysis results are not included in the Flex configuration file, but can be saved separately in an Application Profiler project file by clicking **Save Project** from the main menu.

Select **Load Project** to make the full-fidelity analysis details available again. You can use the filter and optimization settings to generate a Flex configuration file.

The Application Profiler project file can only be opened in Application Profiler and cannot be used in User Environment Manager.

Advanced Configuration of Application Profiler

Use these settings and preferences for advanced configuration of Application Profiler.

Table 3-1. Application Profiler Settings

Setting	Description
Use Deepest Registry Path	<p>If you enable this setting, only the most specific registry paths are kept.</p> <p>This setting is enabled by default.</p>
Support HKLM	<p>If you enable this setting, access to the HKEY_LOCAL_MACHINE registry hive is included in the analysis. If you disable the setting, only HKEY_CURRENT_USER is taken into account.</p> <p>This setting is disabled by default.</p>
Enable Registry Exclusions	<p>Controls whether registry exclusions are applied.</p> <p>This setting is enabled by default.</p>
Enable File Exclusions	<p>Controls whether file system exclusions are applied.</p> <p>This setting is enabled by default.</p>
Show Unsupported File Access	<p>User Environment Manager supports managing file system settings only in the user profile subfolders. Application Profiler filters out any file access outside the user profile subfolders by default. If you enable this setting, access to files outside the user profile is included in the Flex configuration file. FlexEngine cannot process such entries, but this feature can be useful when you analyze an application that is saving its settings in nonstandard locations.</p> <p>This setting is disabled by default.</p>
Warn If Project Not Saved	<p>If you enable this setting, Application Profiler displays a warning message when you start a new profiling session or exit Application Profiler, and the project file for the previous session is not saved.</p> <p>This setting is disabled by default.</p>
Default Save Path	<p>With this setting, you can configure a default location for the Save As dialog box that appears when you save the Flex configuration file. If you configure a default path, the Save As dialog box always opens to the configured folder.</p> <p>This setting is not configured by default.</p>
Save Icon	<p>If you enable this setting, the icon of the analyzed application is saved when you save the Flex configuration file. If the User Environment Manager Management Console finds a .ICO file next to a Flex configuration file with the same name, it uses that icon in the tree.</p> <p>This setting is enabled by default.</p>
Browse Local Profile	<p>If you enable this setting, the Flex Config File editor completes the registry and file system paths with information from the user profile.</p> <p>This setting is enabled by default.</p>
Display Predefined Settings Size	<p>If you enable this setting, and you select Save Config File with Predefined Settings, Application Profiler displays the size of the generated predefined settings archive.</p>

Using File System Auto Excludes

Many applications create log files, temporary files, diagnostic folders, crash dump folders, or other items, that typically do not require that User Environment Manager manage them. These items must not be saved to a user profile archive.

With the File System Auto Excludes feature you can specify folder names and filenames that, when encountered during profiling, generate exclude sections, as the following example shows.

```
[IncludeFolderTrees]
<AppData>\Vendor\Software

[ExcludeFolderTrees]
<AppData>\Vendor\Software\Crash Reports

[ExcludeFiles]
<AppData>\Vendor\Software\*.log
<AppData>\Vendor\Software\*.tmp
```

Without File System Auto Excludes, the complete `<AppData>\Vendor\Software` tree is exported, including crash reports, log files, and temporary files.

If you configure `Crash Reports` as a folder to exclude and `*.log` and `*.tmp` as files to exclude, Application Profiler populates the exclude sections with the appropriate paths.

The File System Auto Excludes setting is applied only if the File System Auto Excludes setting is enabled. Folders and files to be excluded are configured in the Manage Excludes dialog box.

Manage Excludes

In the **Folders to exclude** text box, you can specify partial folder names like `Crash Reports` or `Diagnostics\Crash Reports`, and absolute ones like `<AppData>\Vendor\Software\Crash Reports`.

Path components can contain wildcards, for example `Diag*\Crash Reports`, however, matching takes place component-by-component and folders such as `Diagnostics\Troubleshooting\Crash Reports` do not match in this case.

In the **Files to exclude** text box, you specify entries, such as `*.log` and `*.tmp`. You can only specify filenames, with wildcards supported. Path information is not supported.

If the entries in the Manage Excludes dialog box are not specific enough, they might result in the exclusion of all file system-related configuration.

Command-Line Arguments

Application Profiler supports command-line arguments that you can use to specify configuration settings or to allow profiling an application in an unattended or semiunattended manner.

Unless indicated differently, all command-line arguments are optional.

Table 3-2. Command-Line Arguments for Configuration Settings

Argument	Description
/FileOpt <i>Level</i>	Sets the file system optimization level to the specified value. Passing a level of 0 disables file system optimization.
/RegOpt <i>Level</i>	Sets the registry optimization level to the specified value. Passing a level of 0 disables registry optimization.
/Deep <i>Value</i>	Configures the Use Deepest Registry Path setting. Passing a value of 0 disables the setting, 1 enables it.
/HKLM <i>Value</i>	Configures the Support HKLM setting. Passing a value of 0 disables the setting, 1 enables it.
/FileExcl <i>Value</i>	Configures the Enable File Exclusions setting. Passing a value of 0 disables the setting, 1 enables it.
/RegExcl <i>Value</i>	Configures the Enable Registry Exclusions setting. Passing a value of 0 disables the setting, 1 enables it.
/AutoExcl <i>Value</i>	Configures the Auto Exclude Sections setting. Passing a value of 0 disables the setting, 1 enables it.
/SaveIcon <i>Value</i>	Configures the Save Icon setting. Passing a value of 0 disables the setting, 1 enables it.
/DefaultPath <i>Path</i>	Configures the Default Save Path setting. Passing an empty path disables the setting.
/LoadProject <i>Path</i>	Loads a project.
/Reset	Resets all configuration settings to their default values.

Table 3-3. Command-Line Arguments for Unattended and Semiattended Use

Argument	Description
/EXE <i>Exe</i>	The fully qualified path of the application to analyze. If you use this argument, you must also specify /FlexConfig .
/SelectEXE	Displays a dialog box to select the application to analyze. If you use this argument, you must also specify /FlexConfig .
/FlexConfig <i>Path</i>	The fully qualified path of the Flex configuration file that must be created. If you use this argument, you must also specify /EXE or /SelectEXE .
/StartIn <i>Path</i>	Specifies the start folder of the application.
/Parameter <i>Param</i>	Specifies an argument to the application. You can use the /Parameter argument multiple times.
/PreDef	Indicates that a predefined settings archive must be created based on the generated Flex configuration file.
/AutoCloseTimeout <i>Timeout</i>	Indicates that Application Profiler must close the application after the specified number of seconds.
/AutoCloseRetry <i>RetryCount</i>	Use this argument to set the number of retries if the application does not close after the auto-close timeout. If you use this argument, you must also specify /AutoCloseTimeout .

Table 3-3. Command-Line Arguments for Unattended and Semiattended Use (continued)

Argument	Description
/ShowResult	Without this argument, Application Profiler runs fully unattended. If you specify /ShowResult , the main user interface appears when the profiling session has ended.
/SaveProject <i>Path</i>	Saves a project.