



Application Profiler Administrator's Guide

VMware[®] User Environment Manager

VERSION 8.7.0

You can find the most up-to-date technical documentation on the VMware Web site at:
<http://www.vmware.com/support/>

The VMware Web site also provides the latest product updates.

Copyright © 2015 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

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

Contents

1	Introduction	5
2	Installation	6
2.1	System requirements.....	6
2.2	Installing Application Profiler	7
2.2.1	Manual installation	7
2.2.2	Unattended installation.....	8
2.3	Uninstalling Application Profiler	8
3	Using Application Profiler	9
3.1	Starting a profiling session	9
3.2	Launching the application to be profiled	10
3.3	Changing the application's configuration	11
3.4	Analysis result: a Flex config file	11
4	Application Profiler Options	12
4.1	Filtering and optimizing the analysis details	12
4.1.1	Filtering.....	12
4.1.2	Optimizing	12
4.2	Starting a profiling session	13
4.3	Editing the Flex config file	14
4.4	Saving the Flex config file	15
4.4.1	Predefined settings	15
4.5	Editing a profile archive	15
4.6	Exclusions	16
4.6.1	Registry Exclusions	16
4.6.2	File System Exclusions	16
4.7	Application Profiler project files	16
5	Settings	17
5.1	Use Deepest Registry Path	17
5.2	Support HKLM.....	17
5.3	Enable Registry Exclusions	17
5.4	Enable File Exclusions	17
5.5	Show Unsupported File Access	17
5.6	Warn If Project Not Saved	18
5.7	Default Save Path	18

5.8	Save Icon.....	18
5.9	Browse Local Profile.....	18
5.10	Display Predefined Settings Size.....	18
5.11	File System Auto Excludes	18
5.11.1	Auto Exclude Sections	19
5.11.2	Manage Excludes	19
6	Command-line arguments.....	20

1 Introduction

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

Since the introduction of Immidio Flex Profiles (the predecessor of Immidio Flex+ and VMware User Environment Manager), the only labor-intensive task when implementing it was the creation of Flex config files. With Application Profiler this task has been drastically simplified: just run your application from the user-friendly graphical interface and Application Profiler automatically analyzes where it stores its file and registry configuration. The analysis results in an optimized Flex config file, which can then be edited in the Application Profiler or used as is in the User Environment Manager environment.

Another task that has become much easier is the creation of application-specific *predefined settings*, which allow you to set the initial configuration state of applications. Simply use the Application Profiler to analyze the application, and configure it exactly to your liking before closing it. Save the Flex config file with predefined settings to export the current application configuration state.

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

2 Installation

2.1 System requirements

The following Windows versions and editions are supported by Application Profiler:

- Windows Server 2003 Standard and Enterprise x86 and x64 SP2
- Windows Vista Business, Enterprise and Ultimate x86 and x64 SP2
- Windows Server 2008 Standard and Enterprise x86 and x64 SP2
- Windows 7 Professional, Enterprise and Ultimate x86 and x64 SP1
- Windows Server 2008 R2 Standard and Enterprise x64 SP1
- Windows 8 Professional and Enterprise x86 and x64
- Windows Server 2012 Standard and Datacenter x64
- Windows 8.1 Professional and Enterprise x86 and x64 with update 1
- Windows Server 2012 R2 Standard and Datacenter x64 with update 1
- Windows 10 Professional and Enterprise x86 and x64

Application Profiler requires .NET Framework 2.0 SP1 to be installed. On Windows 8, Windows 8.1, and Windows 10 this entails [installing .NET Framework 3.5](#), as described on the Microsoft website.

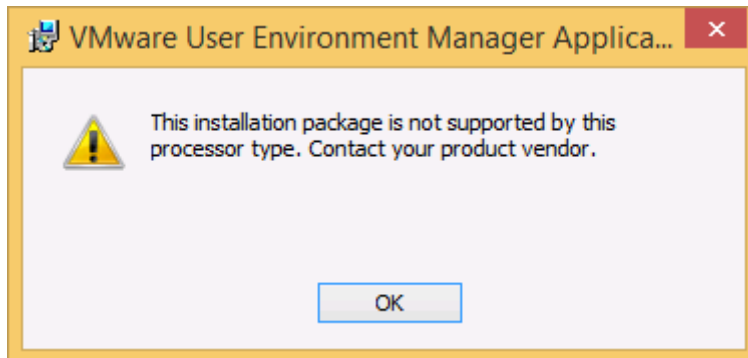
No specific hardware other than the typical operating system requirements and requirements of installed applications is required for Application Profiler.

NOTE: It is strongly recommended to install Application Profiler in a reference environment that corresponds with the environment that VMware User Environment Manager is deployed in. This ensures that the applications to be analyzed behave similarly to the ones used in production.

When using VMware User Environment Manager on a Terminal Server (RDSH), however, it is recommended to install the Application Profiler on the equivalent desktop OS.

2.2 Installing Application Profiler

The Application Profiler MSI is available in two editions – one for 32-bit systems, and one for 64-bit. If you try to install an MSI that does not match the architecture of your OS, an error message will be displayed:



To resolve this issue, use the download package that matches your OS architecture.

NOTE: The MSI file has a digital signature, which the Windows Installer infrastructure will validate when the installation is started. This includes a so-called certificate revocation check, for which the system needs Internet access. If there is no (sufficient) Internet connectivity, the installation will continue, but only after several timeouts – in the meantime, the installer seems to stall without providing any feedback.

2.2.1 Manual installation

Installing Application Profiler is very straightforward:

1. Run *VMware UEM Application Profiler 8.7 x86.msi* or *VMware UEM Application Profiler 8.7 x64.msi* after uncompressing the download package.

IMPORTANT: The MSI file must be executed under an account with administrative privileges.

2. The *Application Profiler Setup Wizard* starts and displays the *Welcome* page. Click *Next*.
3. The *License Agreement* page is displayed. Please read it carefully by scrolling down the license text. You must accept the license agreement before you can click *Next*.
4. The next page lets you select the destination folder. To install into a different folder, click the *Browse* button and select another folder. Click *Next*.
5. The *Ready to install* page is displayed. Proceed by clicking *Install*.
6. The installation starts.
7. When the installation is finished, a final page shows the success. Click *Finish* to exit the installation.

2.2.2 Unattended installation

In most enterprise environments, Electronic Software Distribution systems are used to install software in an automated fashion. The Application Profiler MSI supports unattended installation, using the following MSI property to select the installation directory.

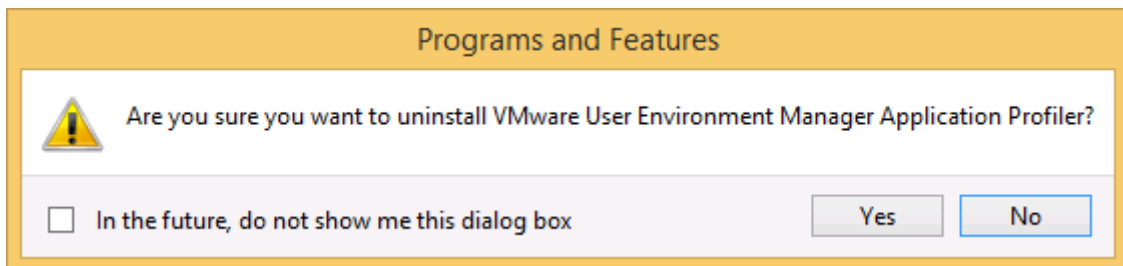
Property	Description
INSTALLDIR	The absolute path to the installation directory. Defaults to %ProgramFiles%\Immidio\Application Profiler.

An example of a custom unattended installation command (split across multiple lines for readability only):

```
msiexec.exe /i "VMware UEM Application Profiler 8.7 x86.msi" /qn
INSTALLDIR="D:\Apps\VMware UEM Application Profiler"
/l* InstallProfiler.log
```

2.3 Uninstalling Application Profiler

You may uninstall Application Profiler from your computer by opening *Add or Remove Programs* (in Windows XP and Windows Server 2003) or *Programs and Features* (in newer Windows versions) in the *Control Panel*. Selecting Application Profiler and clicking the *Remove* or *Uninstall* button opens a wizard which guides you through the steps to remove the software successfully.



3 Using Application Profiler

As described in chapter 1, Application Profiler analyzes at which registry and file system locations the settings for a particular application are stored, to create a Flex config file for use with VMware User Environment Manager.

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

1. Start Application Profiler.
2. From within Application Profiler, launch the application that you wish to analyze. In the background, Application Profiler will monitor the registry and file system actions of the launched application.
3. Change some settings in the application, to make sure that all application settings will be saved, and then exit the application.
4. Application Profiler now stops monitoring and outputs the collected information as a Flex config file.

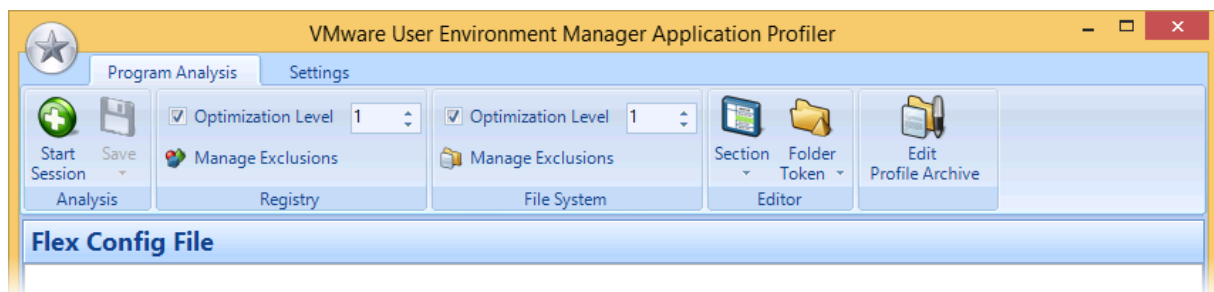
This chapter further describes these steps. The next chapter goes into all options in more detail.

3.1 Starting a profiling session

As described in section 2.1, it is recommended to run Application Profiler on a system that corresponds to the environment that VMware User Environment Manager is used in. In this guide, we will refer to that system as the *profiling system*.

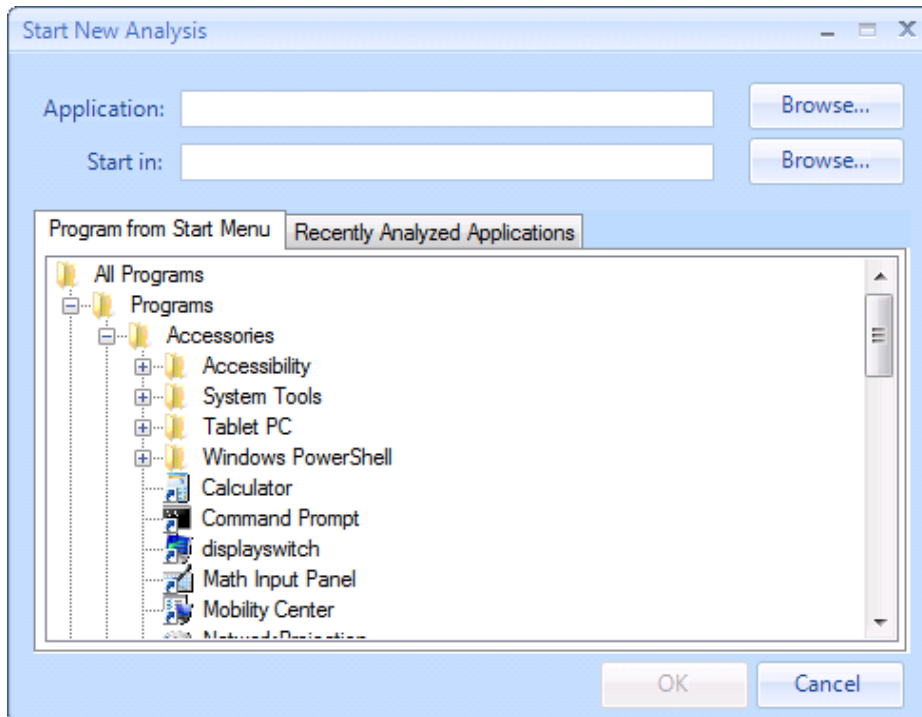
To create a Flex config file for an application, install that application on your profiling system. Then log on to your profiling system using an administrator account and start Application Profiler.

The main interface appears:



3.2 Launching the application to be profiled

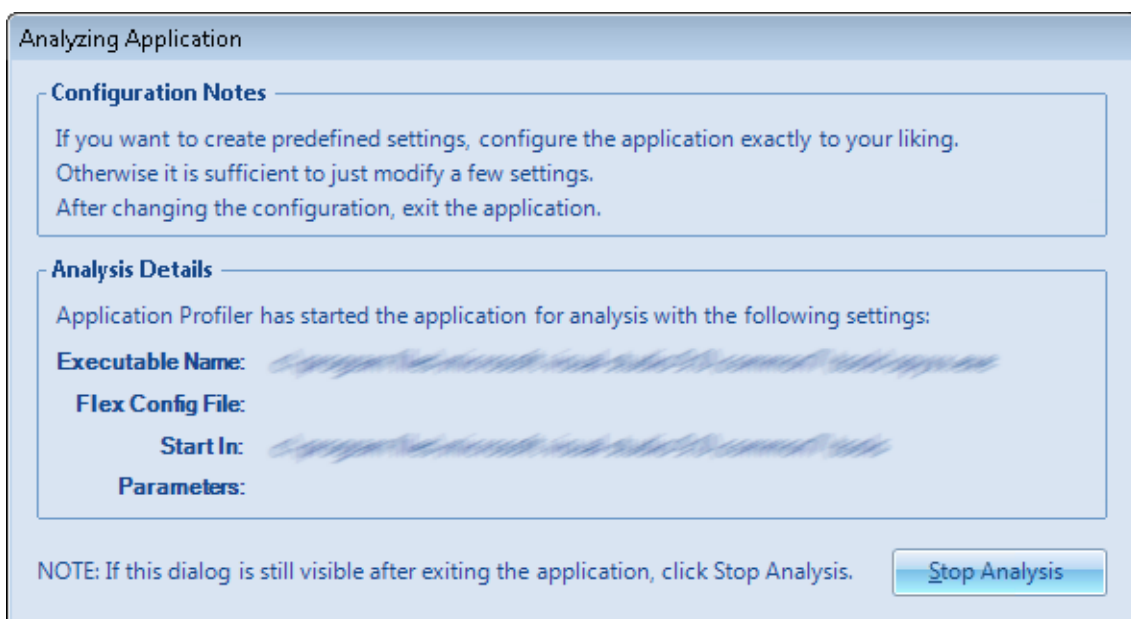
Click the  *Start Session* button. The *Start New Analysis* dialog appears:



Here you can specify the application you want to create a Flex config file for, either by selecting it from the *All Programs* tree, or by browsing to the application's executable.

You can optionally enter command line arguments after the application's path in the *Application* text box (just be sure to put quotes around the path if necessary), and specify the application's start folder in the *Start in* textbox. If you select a shortcut from the *All Programs* tree, all settings are read from the shortcut.

Click *OK* to have Application Profiler launch the application. The main user interface disappears, and the *Analyzing Application* dialog is displayed:



3.3 Changing the application's configuration

Application Profiler monitors the changes that the application makes to the registry and the file system, so during the analysis session it is important to make the application save its configuration.

Typically it is sufficient to just modify a few of the settings – many applications save their full configuration whenever a change is made. Sometimes you might need to touch a few more of the application's features and settings, like creating a signature in an email client or adding an entry to the custom dictionary in a word processor, so the corresponding files are written to disk.

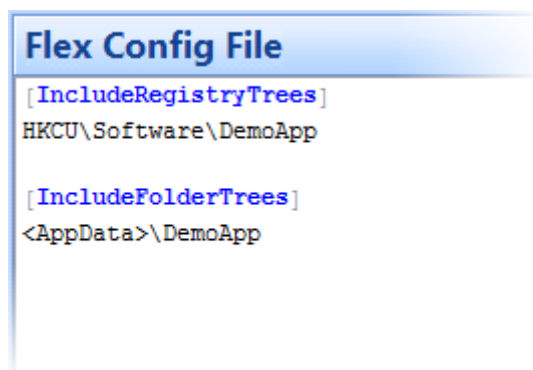
NOTE: To just find the registry and file system definitions for a Flex config file, the actual configuration settings are not relevant. Only the *locations* where the settings are kept in the user profile are important.

If you also want to create *predefined settings*, however, the actual configuration settings do matter, as they will become part of the predefined settings archive (cf. section 4.4.1).

When you are done making changes, exit the application.

3.4 Analysis result: a Flex config file

After you exit the application, the main Application Profiler user interface reappears, with the analysis results:




```

Flex Config File
[IncludeRegistryTrees]
HKCU\Software\DemoApp

[IncludeFolderTrees]
<AppData>\DemoApp
  
```

You can adjust these generated settings by modifying some of the Application Profiler options or making manual changes in the editor. This is described in detail in chapter 4.

Once the Flex config file is to your liking, you can  Save it.

NOTE: Application Profiler monitors the application you selected, and all the child processes started by that application. Monitoring stops automatically when the main application and all child processes have exited.

For some applications, one or more of those child processes continue to run even when you exit the main application. In that case, you will have to manually stop the analysis by clicking the *Stop Analysis* button in the *Analyzing Application* dialog. Note that this will not abort the child processes – it only affects the monitoring by Application Profiler.

4 Application Profiler Options

The previous chapter gave a high-level overview of the Application Profiler functionality. This chapter describes in more detail how the information collected during the analysis is processed, and explains all the program's features and settings.

4.1 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 config file is generated, using a number of optimizations and filtering options.

NOTE: The full-fidelity analysis details are available until you start a new profiling session or exit Application Profiler. If you wish, you can save them to disk using the *Save Project* menu option as described in section 4.7.

4.1.1 Filtering

By default, file system access to locations outside the user profile is filtered out, because VMware User Environment Manager does not support managing files outside the user profile. This behavior can be changed by enabling the *Show Unsupported File Access* setting (cf. section 5.5).

Similarly, access to the HKEY_LOCAL_MACHINE registry hive is excluded unless the *Support HKLM* setting (cf. section 5.2) is enabled. Note that VMware User Environment Manager only supports managing HKLM registry information when used in conjunction with application virtualization.

After this initial filtering, the so-called *Exclusions* are applied. For both the registry and the file system a number of paths are defined by default, to exclude settings that are typically not application-specific. These exclusions can be edited via *Manage Exclusions* (cf. section 4.6.1 for the registry, and 4.6.2 for the file system). Whether the exclusions are in effect is controlled by the *Enable Registry Exclusions* and *Enable File Exclusions* settings (cf. sections 5.3 and 5.4, respectively).

As the final filtering step, the *Use Deepest Registry Path* setting (cf. section 5.1) is taken into account. If enabled, only the "deepest" (i.e. most specific) registry paths are kept, and all intermediate paths are filtered out, as they will be recreated anyway because of the "deep" path.

4.1.2 Optimizing

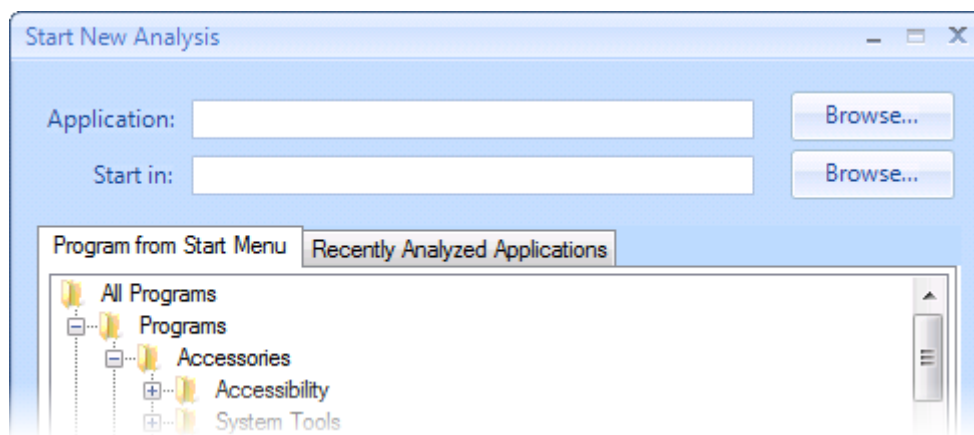
After the filters have been applied, an optimization step can be performed. For the registry, this means that access to separate registry keys and values below a common registry key `HKCU\Software\vendor` are "rolled-up" to a single reference to the registry tree `HKCU\Software\vendor`.

For the file system, a similar optimization applies to file and folder access below a common `<AppData>\vendor` or `<LocalAppData>\vendor` folder.

The *Optimization Level* controls how many key or folder levels remain after optimization. By default, both are set to 1, resulting in paths like `HKCU\Software\vendor` and `<AppData>\vendor`. If you want to separately manage multiple applications by the same vendor where configuration is stored in `...\vendor\Application\...` paths, the level should be increased to 2, to make each Flex config file only manage the settings for a particular application.

4.2 Starting a profiling session

Click the  *Start Session* button. The *Start New Analysis* dialog appears:



Here you can specify the application you want to create a Flex config file for, either by selecting it from the *All Programs* tree, selecting a previously analyzed application from the *Recently Analyzed Applications* list, or by browsing to the application's executable.

You can optionally enter command line arguments after the application's path in the *Application* text box (just be sure to put quotes around the path if necessary), and specify the application's start folder in the *Start in* textbox. If you select a shortcut from the *All Programs* tree, all settings are read from the shortcut.

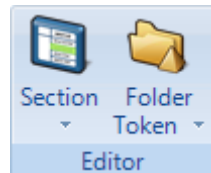
Only executables (`.EXE` files) are supported for analysis. Some applications install shortcuts in the start menu that refer to an application document instead of to the application's executable. You can profile these applications by browsing to the executable and adding any additional arguments after the application's path, as described above.

NOTE: Applications for which an *Application Template* is provided in the VMware User Environment Manager Management Console cannot be selected for analysis. For these applications you can create a Flex config file in the Management Console with the appropriate template.

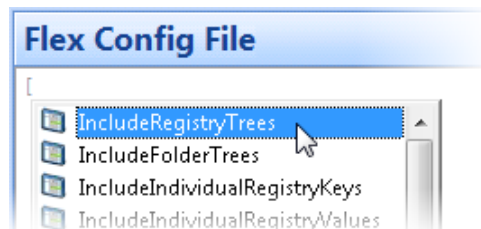
4.3 Editing the Flex config file

The Flex config file content that is generated can be edited, similar to the editor in the User Environment Manager Management Console. Please refer to the *VMware User Environment Manager Administrator's Guide* for a full description of the format of Flex config files.

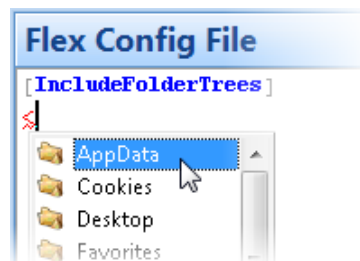
Section headers and folder tokens can be inserted using the corresponding dropdowns in the ribbon:



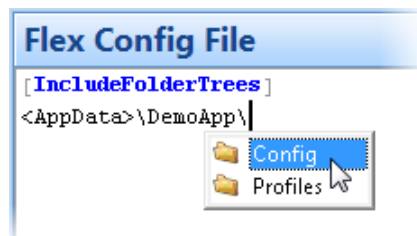
Alternatively, to insert a section header you can type a [(open square bracket) character in the editor, to display a dropdown from which you can select a section header:




When you type a < (smaller than) character in the editor, a dropdown appears from which you can select a folder token:



If the *Browse Local Profile* setting (cf. section 5.9) is enabled, file system and registry path are auto-completed base on the actual contents of the user profile:



4.4 Saving the Flex config file

To save the Flex config file, click  *Save*. If you haven't saved the file before, a *Save As* dialog appears – enter a file name for your Flex config file, and click the *Save* button.

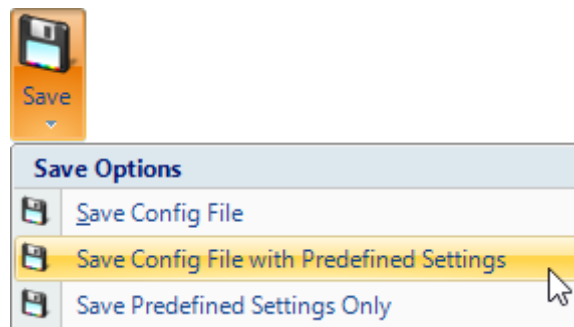
If a *Default Save Path* (cf. section 5.7) has been configured, the *Save As* dialog always opens to that path. If the *Save Icon* setting (cf. section 5.8) has been enabled, the icon of the analyzed application will be saved next to the Flex config file.

The application you started in the profiling session will be included in the Flex config file as a DirectFlex executable and DirectFlex will be enabled. Note that this does not apply to App-V 4.x applications; for these you must manually add integration in the User Environment Manager Management Console (see the *VMware User Environment Manager Administrator's Guide* for details).

NOTE: It is **strongly recommended** to save the Flex config files you create using Application Profiler to a test environment, so you can first make sure that all settings are correctly managed before deploying the configuration files on your production systems.

4.4.1 Predefined settings

If, during analysis, you have configured the application exactly to your liking, you can create predefined settings while saving your Flex config file:




This creates a profile archive to use for pre-populating settings when a user logs on. See the *VMware User Environment Manager Administrator's Guide* for more information about predefined settings.

If the *Display Predefined Settings Size* setting is enabled (cf. section 5.10), the size of the generated profile archive will be displayed afterwards.

You can also create predefined settings without saving the Flex config file itself by selecting the *Save Predefined Settings Only* menu item.

4.5 Editing a profile archive


If you want to check exactly which settings were saved as predefined settings, or want to modify them, you can use  *Edit Profile Archive*. After you have selected the profile archive (.zip file) containing the predefined settings, Windows Explorer will open to a folder containing the extracted settings. For more information, please refer to the *VMware User Environment Manager Administrator's Guide*.

4.6 Exclusions


By default, a number of paths are configured here that typically do not contain application-specific configuration settings and therefore shouldn't be included in the contents of a Flex config file.

You can modify these default settings and add your own exclusions. Exclusions are processed case-insensitively, and can contain * wildcards.

4.6.1 Registry Exclusions

With  *Manage Exclusions* in the *Registry* ribbon group, you can configure which registry paths should always be filtered out. These exclusions are only applied when the *Enable Registry Exclusions* setting is enabled (cf. section 5.3).

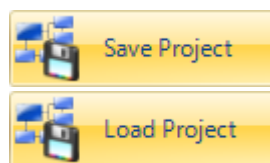
4.6.2 File System Exclusions

With  *Manage Exclusions* in the *File System* ribbon group, you can configure which file system paths should always be filtered out. These exclusions are only applied when the *Enable File Exclusions* setting is enabled (cf. section 5.4).

4.7 Application Profiler project files

After Application Profiler has analyzed an application, the analysis results are filtered and optimized (cf. section 4.1) and a Flex config 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 config file itself, but can be saved separately in a so-called *Project* file. To do so, use the *Save Project* menu option.



The *Load Project* menu option makes the full-fidelity analysis details available again, meaning that you can use the filter and optimization settings to generate a Flex config file.

NOTE: Project files are only used by Application Profiler; you don't need them for VMware User Environment Manager.

5 Settings

5.1 Use Deepest Registry Path

If *Use Deepest Registry Path* is enabled, only the “deepest” (i.e. most specific) registry paths are kept, and all intermediate paths are filtered out, as they will be recreated anyway because of the “deep” path.

This setting is enabled by default.

5.2 Support HKLM

If *Support HKLM* is enabled, access to the HKEY_LOCAL_MACHINE registry hive is included in the analysis. Otherwise, only HKEY_CURRENT_USER is taken into account.

This setting is disabled by default.

5.3 Enable Registry Exclusions

The *Enable Registry Exclusions* setting controls whether registry exclusions are applied (cf. section 4.6.1).

This setting is enabled by default.

5.4 Enable File Exclusions

The *Enable File Exclusions* setting controls whether file system exclusions are applied (cf. section 4.6.2).

This setting is enabled by default.

5.5 Show Unsupported File Access

VMware User Environment Manager only supports managing file system settings in the user profile subfolders, so Application Profiler filters out any file access outside the user profile subfolders by default.

If the *Show Unsupported File Access* setting is enabled, access to files outside the user profile is included in the Flex config file. Such entries cannot be processed by FlexEngine, but this feature can prove useful when analyzing an application that is saving its settings in non-standard locations.


This setting is disabled by default.

5.6 Warn If Project Not Saved

If the *Warn If Project Not Saved* setting is enabled, Application Profiler will display a warning message if a new profiling session is started or Application Profiler is exited while the project file for the previous session has not been saved yet (cf. 4.7).

This setting is disabled by default.

5.7 Default Save Path

With the  *Default Save Path* setting, you can configure a default location for the *Save As* dialog that appears when you save the Flex config file.

If a default path is configured, the *Save As* dialog will always open to the configured folder.

This setting is not configured by default.

5.8 Save Icon

If *Save Icon* is enabled, the icon of the analyzed application will be saved as well when you save the Flex config file. If the User Environment Manager Management Console finds a .ICO file next to a Flex config file with the same name, it will use that icon in the tree.

This setting is enabled by default.

5.9 Browse Local Profile

If *Browse Local Profile* is enabled, the *Flex Config File* editor will auto-complete registry and file system paths with information from the user profile.

This setting is enabled by default.

5.10 Display Predefined Settings Size

If the *Display Predefined Settings Size* setting is enabled, and *Save Config File with Predefined Settings* is selected (cf. 4.4.1), Application Profiler will display the size of the generated predefined settings archive.

5.11 File System Auto Excludes

Many applications create log files, temporary files, diagnostic folders, crash dump folders, or other items that typically do not need to be managed using VMware User Environment Manager and therefore should not be saved into a user's profile archive.

The *File System Auto Excludes* feature allows you to specify folder and file names that, when encountered during profiling, will result in the generation of so-called exclude sections:

```
[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 would be exported, including crash reports, log files and temporary files.

However, if **Crash Reports** is configured as a folder to exclude, and ***.log** and ***.tmp** as files to exclude, Application Profiler automatically populates the two **Exclude** sections with the appropriate paths.

NOTE: *File System Auto Excludes* is an advanced feature, aimed at experienced UEM administrators with a thorough understanding of the exclude sections in Flex config files.

The *File System Auto Excludes* logic is only applied if the *Auto Exclude Sections* setting is enabled (cf. 5.11.1).

Folders and files that should be excluded are configured in the *Manage Excludes* dialog (cf. 5.11.2).

5.11.1 Auto Exclude Sections

The *Auto Exclude Sections* setting controls whether the *File System Auto Excludes* logic is enabled (cf. section 5.11).

This setting is disabled by default.

5.11.2 Manage Excludes

In the *Folders to exclude* field, you can specify “partial” folder names like **Crash Reports** or **Diagnostics\Crash Reports**, and absolute ones like `<AppData>\Vendor\Software\Crash Reports` (although you will typically want to configure partial and therefore more generically appropriate paths).

Path components can contain wildcards (e.g. **Diag*\Crash Reports**) but please note that matching takes place component-by-component (so the example given will not match a folder like **Diagnostics\Troubleshooting\Crash Reports**, for instance).

In the *Files to exclude* field, you will typically specify entries like ***.log** and ***.tmp**. You can only specify file names (with wildcards); path information is not supported.

NOTE: If the entries in the *Manage Excludes* dialog are too “broad”, they may result in the exclusion of *all* file system-related configuration. In that case, Application Profiler displays the warning message “*All file system entries were excluded because of Auto Excludes.*”

6 Command-line arguments

Application Profiler supports a number of command-line arguments. Some of these can be used to specify configuration settings, while others allow profiling an application in a (semi-)unattended fashion.

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

Argument	Description
/FileOpt <i>Level</i>	Sets the file system optimization level (cf. section 4.1.2) to the specified value. Passing a <i>Level</i> of 0 disables file system optimization.
/RegOpt <i>Level</i>	Sets the registry optimization level (cf. section 4.1.2) to the specified value. Passing a <i>Level</i> of 0 disables registry optimization.
/Deep <i>Value</i>	Configures the <i>Use Deepest Registry Path</i> setting (cf. section 5.1). Passing a <i>Value</i> of 0 turns the setting off; 1 turns it on.
/HKLM <i>Value</i>	Configures the <i>Support HKLM</i> setting (cf. section 5.2). Passing a <i>Value</i> of 0 turns the setting off; 1 turns it on.
/FileExcl <i>Value</i>	Configures the <i>Enable File Exclusions</i> setting (cf. section 5.4). Passing a <i>Value</i> of 0 turns the setting off; 1 turns it on.
/RegExcl <i>Value</i>	Configures the <i>Enable Registry Exclusions</i> setting (cf. section 5.3). Passing a <i>Value</i> of 0 turns the setting off; 1 turns it on.
/AutoExcl <i>Value</i>	Configures the <i>Auto Exclude Sections</i> setting (cf. section 5.11.1). Passing a <i>Value</i> of 0 turns the setting off; 1 turns it on.
/SaveIcon <i>Value</i>	Configures the <i>Save Icon</i> setting (cf. section 5.8). Passing a <i>Value</i> of 0 turns the setting off; 1 turns it on.
/DefaultPath <i>Path</i>	Configures the <i>Default Save Path</i> setting (cf. section 5.7). Passing an empty <i>Path</i> ("") turns the setting off.
/LoadProject <i>Path</i>	Load a project (cf. section 4.7).
/Reset	Resets all configuration settings to their default values.
<i>Command-line arguments for (semi-)unattended use:</i>	
/EXE <i>Exe</i>	The fully-qualified path of the application to analyze. If this argument is used, /FlexConfig must also be specified.
/FlexConfig <i>Path</i>	The fully-qualified path of the Flex config file that should be created. If this argument is used, /EXE must also be specified.
/StartIn <i>Path</i>	Specifies the application's start folder.

Argument	Description
/Parameter <i>Param</i>	Specifies an argument to the application. The /Parameter argument can be used multiple times.
/PreDef	Indicates that a predefined settings archive must be created based on the generated Flex config file.
/AutoCloseTimeout <i>Timeout</i>	Indicates that the application must be automatically closed after the specified number of seconds.
/AutoCloseRetry <i>RetryCount</i>	Use this argument to specify the number of retries if the application does not close after the auto-close timeout. If this argument is used, /AutoCloseTimeout must also be specified.
/ShowResult	Without this argument, Application Profiler runs fully unattended. If /ShowResult is specified, the main user interface appears when the profiling session has ended.
/SaveProject <i>Path</i>	Saves a project (cf. section 4.7).