CAMEYO

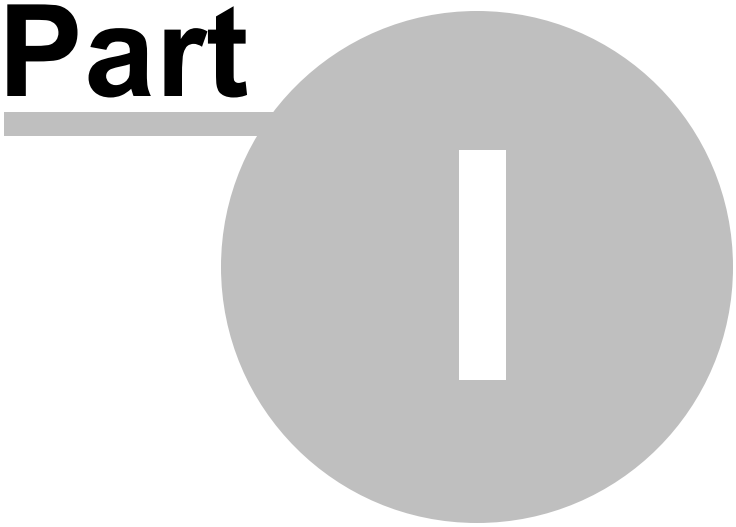
USER GUIDE

Because Software should be used, not installed

VERSION 2.5

Table of Contents

Cameyo basic usage



1 Cameyo basic usage

1.1 Packaging your own virtual app

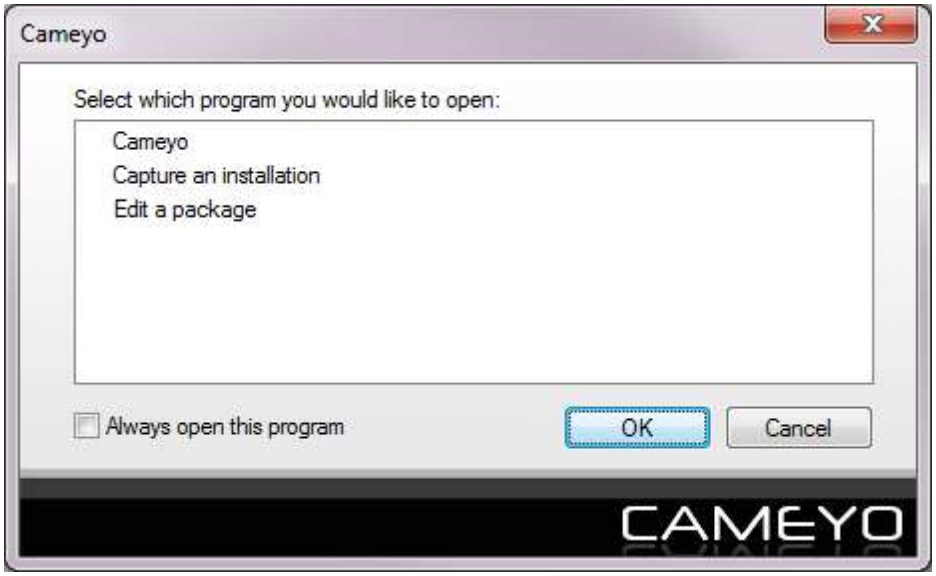
The process of turning a software into a virtual package is called "packaging". The process consists of installing a software while Cameyo "captures" it. Once this step is done, you will be able to use this software on any machine without having to re-install it again. Also, virtual software can work in "isolated mode", never interfering with your Windows stability (unlike regular software).

Cameyo is arguably the simplest application virtualization software for creating and editing virtual software packages. Just follow the below simple steps.

Note: packaging has to be done on a clean virtual machine. Avoid packaging on your own machine or any machine which has other software & components installed on it.

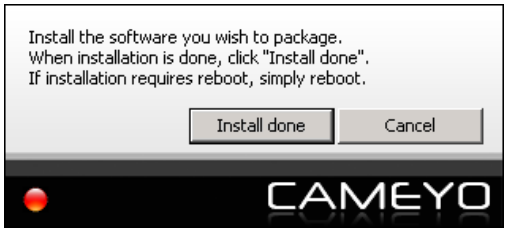
1. Launch Cameyo's Packager:

Launch Cameyo and select "Capture an installation" in the startup menu:

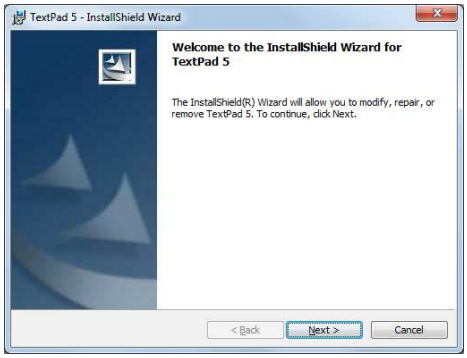


2. Wait for the Packager to capture a pre-installation snapshot of your system in its current state.

3. Once the snapshot is taken, the following window will appear:



At this point, install the software you wish to package:

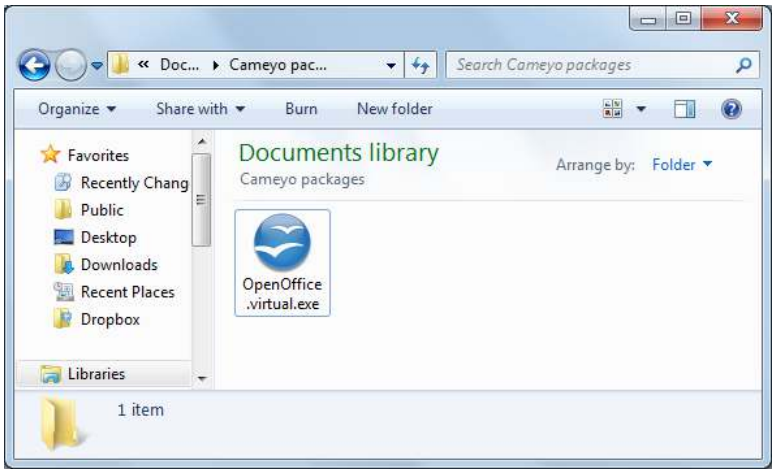


(Random installation screenshot)

Configure your application the way you want it to run.

4. When your software finished installing, press "Install done" on Cameyo's packager window. Cameyo will again take a few minutes to establish a snapshot of your system (this time post-installation) and analyze the differences.

5. You're done! The package is placed under the "My documents\Cameyo packages" folder:



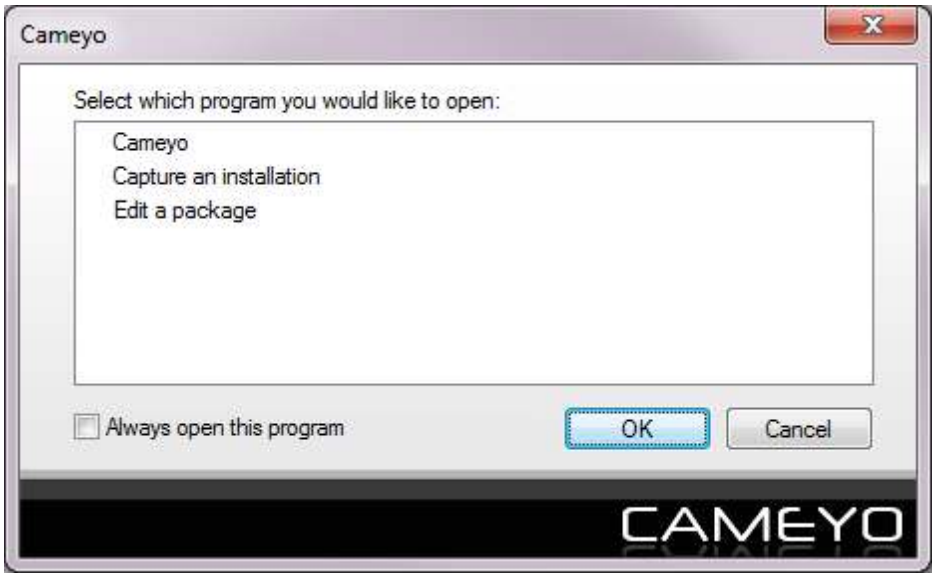
Cameyo automatically finds your application's shortcuts, icon, name, auto launch programs etc. Still, you can customize the package if you wish (see next chapter).

1.2 Editing a virtual package

Unlike other application virtualization products, Cameyo does not require you to edit every virtual package you create. Still, if you wish to customize or optimize your package, you can.

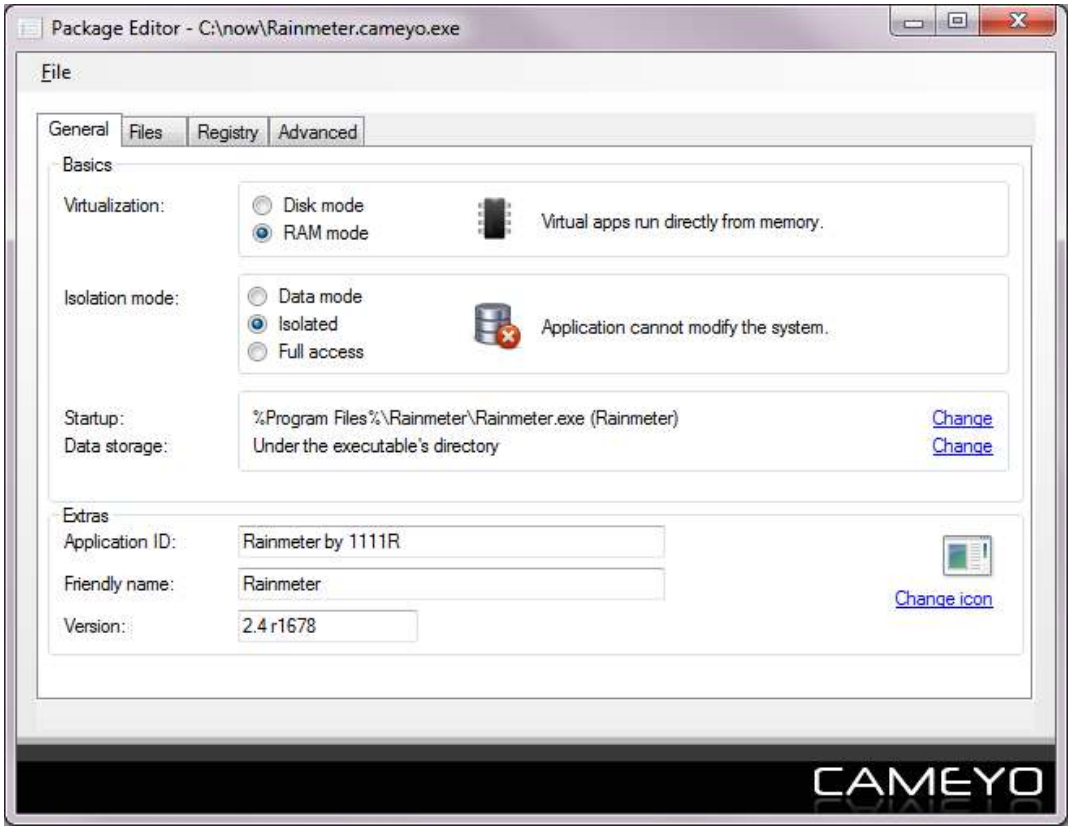
You can edit a package either by dragging it into the Cameyo.exe executable:

OR by launching Cameyo and selecting "Edit a package":



... and then opening the package file to edit (a .cameyo.exe file), either by clicking on File | Open or by dragging & dropping the package file into the window.

Once the package opened, the screen will look as follows:



Basics:

- Virtualization: Cameyo allows two different virtualization modes: Disk and RAM.
  - Disk mode: extracts the application's files on disk as they are needed, and provides them to the operating system. This mode occupies more hard-disk space but is faster for applications which are frequently used. During the very first application's launch, it takes some time for the directory structure to initialize.
  - RAM mode: emulates the application's files and programs directly from memory, and extracts very few files on disk. This mode occupies less hard-disk space.
- Isolation mode: controls whether the virtual application can modify file / registry entries on the target system or not.
  - Isolated (default): the virtual application will not be able to change any files or registry entries on the target machine. Instead, any modification it attempts to do will be redirected to a separate folder or registry key. That folder can be configured by the "Data storage" entry.
  - Full access: the virtual application will be able to change files and registry entries on the target machine.
- Startup: controls which program is to be launched upon execution of the virtual package. If your application only has one possible executable, then a direct launch will be selected by default. Otherwise, a menu can be displayed.
- Data storage: controls where the virtual application places its cache files, and also the isolated changes the application performs.
  - Use either hard disk or USB key, wherever application is launched from (default): if the virtual application's exe is launched from a removable media, the data storage path will be a new directory beneath the executable's directory. Otherwise, if it is launched from the local hard disk

(or from an Internet download), it will be stored under the system's Application Data folder.
    - Under the executable's directory: the data storage path will be a new directory beneath the executable's directory.
    - Custom location: any location you wish to specify. See the chapter Path variables. In addition to these variables, a special variable is available for this setting only: %ExeDir%, which means: the executable's directory.
- Icon: allows you to visualize or change the icon of the virtual package itself (the .cameyo.exe file).

Extras:

- Application ID: this is the ID used internally by Cameyo for uniquely identifying the application. It is also the name of the sub-directory into which Cameyo will expand the application's local storage files.
- Description: any arbitrary description for the package. This is for your own use, and does not affect any of Cameyo's behavior.

Files:

Here you can edit the file system contained (virtualized) inside the virtual package. These are the files as the virtual application will see them. For the virtual application, these will be combined with the other directories & files on the target machine. Here are the operations you can perform on files:

- Add file: add additional files into the package. The files will be added into the folder you highlight.
- Remove file: remove unneeded files or folders from the package.
- Add empty folder: adds a folder with any name you choose. This can be helpful for example if you want to add a folder with different isolation properties (such as a "shared folder" without isolation).
- Save file as: extracts a specific file from the package to your hard drive.
- Isolation (Isolated / Full access): modifies the isolation properties for a specific folder. Folders for which isolation is turned on are shown with an "x" on their folder icons. Setting the isolation folder for the top-most node ("FileSystem") controls the isolation properties for the entire system (not just the folders displayed in the Files editor).
- [x.xx MB]: displays the total size of the highlighted folder, including its sub-folders.

Registry:

Displays the registry as simulated (virtualized) to the virtual application. These are the registry entries as the virtual application will see them. For the virtual application, these will be combined with the other registry entries on the target machine.

As of now, this window only allows you to view the registry entries and control their isolation properties. More editable options will be included into Cameyo in future versions.

Advanced:

- Integration: controls the way the virtual package is integrated or not into Windows:
    - No integration (default): the virtual application will end when it's closed.
    - Recreate shortcuts and associations: the virtual application will recreate the application's shortcuts and associations when it runs. For example, if an application would normally produce shortcuts in the Start menu and the desktop, they will be recreated when the virtual application is first run, and removed when then virtual application is removed from the system. Also, if the application would normally associate with certain files (say: .PDF files), those file types will be associated when the virtual application is first run, and removed when it is removed. To control this feature in more detail, see Virtual package command lines.

    - Virtual integration: this option makes the virtual application inject a non-persistent virtual layer into Windows Explorer. By doing so, the Windows shell looks and feels like the application was truly installed. Windows Explorer will show the application's shortcuts, associations, context menu, and will even show the folders and files of the application as if it was truly installed. To control this feature in more detail, see Virtual package command lines.
- Remove traces upon app exit: allows deleting traces left by the virtual application when it quits. It is important to understand that while Cameyo's virtual apps leave some internal registry keys and files in a separate place on your machine (called "VOS"), those do not affect the behavior of your system in any way. In fact, Windows does not look into the VOS folders. Cleanup takes place when the last process of an application has finished execution.
    - No cleanup, leave app ready to run (default, fastest): the virtual application's traces are not removed until the virtual application is removed from the system.
    - Leave no registry traces: temporary registry keys are removed each time the virtual application quits. Note that the application's registry changes remain persistent (will remain the next time you run the application), because Cameyo saves them into a file named "VirtRegSync.export".
    - Remove all virtual data upon exit: if you don't need the application's data to remain persistent, and you're planning to use this application on a public / shared computer, this option may be useful for you. It leaves no traces each time the application quits.
    - Ask for confirmation before removal: you will be asked for confirmation for the above cleanup options, each time the application quits.
- Expiration: defines an expiration date for the virtual application to stop working.
- Custom events: allows you to define custom events that run when the virtual application starts or stops. See Custom events and scripting.
- Exclude child processes from virtualization: here you can define names of child processes (with or without paths, variablized or not), to run outside of the virtualization context, when they are executed by the virtual application. For example, if your virtual application may run Microsoft Word as a child process and you want to exclude it, you can enter here "winword.exe". If you would like to add Microsoft Excel to this exclusion, you'd put: "winword.exe;excel.exe".

1.3 Using virtual packages

You can launch your virtual package (.cameyo.exe) file on any computer. Upon first execution, it will pre-load the package and extract the basic files needed for the package. In "Disk virtualization mode", the first execution is always somewhat slower.

If your package is running in "Isolated" mode (default), it cannot destabilize your system or leave any traces, and can be completely removed at any time using the "-Remove" command.

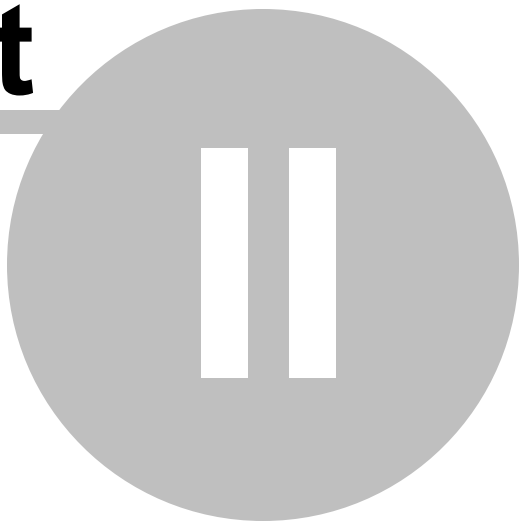
To remove a virtual application, just type the path of its executable followed by "-Remove". For example:

"C:\My Packages\OpenOffice.cameyo.exe" -Remove

Note: virtual applications some times do not behave as expected. If you encounter problems, you are invited to submit them on Cameyo's Forum in the section "Virtual applications not working well".

Package maintenance

Part



2 Package maintenance

2.1 Updating and patching

At times you may want to update virtual packages with new software versions or patches.

Let's distinguish between different types of updates:

1. Major update (i.e. from version 1.x to version 2.x of a software).
2. Minor update / security patch.
3. Single file replacement.

There are different ways to update Cameyo packages:

- Re-packaging the latest version: you can always re-capture the latest version of the software. As long as the package keeps the same AppID, running the package will automatically patch any existing binaries and registry keys before running, without affecting user-created data files. This method works for all update types.
- Using the -Patch parameter on the virtual package (see Virtual package command lines). This method is mostly recommended for update type #2.
- Adding / replacing specific files through the Package Editor's File manager. This method is mostly recommended for update type #3.

2.2 Upgrading to a newer virtualization engine

When a newer Cameyo version is released, you may want to upgrade your existing packages to the new virtualization engine version. This is usually not necessary if the package functions properly, but sometimes Cameyo bugfixes or features may motivate you to upgrade your existing packages.

There are several ways of doing this, depending on how much you need to automate the process and how old the virtual package is:

Quick methods (package version >= 2.5.1137)

- **Drag & drop:**

  Simply drag the new Cameyo version (i.e. Cameyo.exe) in Windows Explorer, and drop it on the virtual package's executable.

- **Quick command-line:**

Run your new Cameyo version (i.e. Cameyo.exe) from the command-line with parameter -UpgradePkg:

Cameyo.exe -UpgradePkg "C:\My Apps\App.cameyo.exe" "C:\New Cameyo Version.exe"

Where:

- "C:\My Apps\App.cameyo.exe" is to be replaced with the virtual package to be upgraded.
- "C:\New Cameyo Version.exe" is the new Cameyo version, whose engine will be used.

Can be combined with the -quiet parameter for full automation (exit code 0 means success):

Cameyo.exe -Quiet -UpgradePkg "C:\My Apps\App.cameyo.exe" "C:\New Cameyo Version.exe"

Technical / legacy method (package version >= 1.6)

- **-ChangeLoader and -ChangeEngine:**

Find Cameyo's Packager.exe within your VOS directory. It should be in directory C:\Users\[username]\AppData\Roaming\VOS\Cameyo\PROG\%Program Files%\Cameyo

Run Packager.exe with parameter -ChangeLoader [app_exe] [new_cameyo_loader_exe]. This will upgrade the virtualization loader. The resulting file will end with a ".new.exe" suffix.

Run Packager.exe with parameter -ChangeEngine [app_exe] [new_cameyo_appvirtdll_dll] (remember that app_exe now ends with a ".new.exe" suffix).

The Loader.exe and AppVirtDll.dll are both within Cameyo's directory (the same directory as Packager.exe).

For example, for a package named "C:\App.cameyo.exe", follow the following sequence from a command line:

1. cd "\Users\[username]\AppData\Roaming\VOS\Cameyo\PROG\%Program Files%\Cameyo"
2. Packager.exe -ChangeLoader "C:\App.cameyo.exe" .\Loader.exe
3. Packager.exe -ChangeEngine "C:\App.cameyo.new.exe" .\AppVirtDll.dll
4. The resulting upgraded package will be in "C:\App.cameyo.new.exe"

Steps 2 and 3 can be fully automated by adding the "-quiet" parameter right after Packager.exe (exit code 0 means success).

2.3 Keeping app settings across different machines

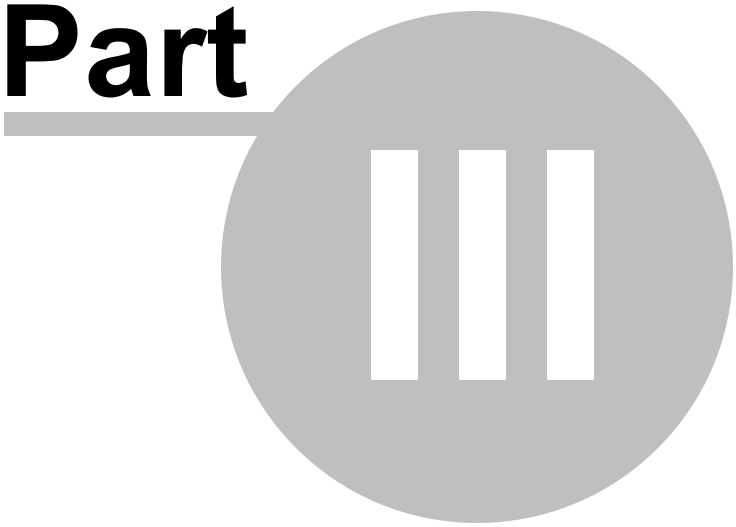
You can keep certain settings within an application and take them with you wherever you go.

This can be done in a number of ways:

- When initially installing & packaging the software, any settings you modify before ending the capture, will be kept within the application. Example:
- Start the application capture process.
- Install the app you wish to package.
- Run your app and configure it however you wish.
- Finish the capture process.
- While using the app, you can modify your settings and then repackage the virtual app. Example:
- Use your app, configure it as you wish, and close it.

- Run your app with the "-Repackage" parameter (i.e. "MyApp.cameyo.exe -Repackage").
- By putting your virtual app in your Dropbox folder, its settings are synchronized among computers as well. Example:
- Move your app into your Dropbox folder.
- Use it.
- Whenever you go to another machine that has access to your Dropbox account, the savings will be shared as well.

Advanced topics



3 Advanced topics

3.1 Virtual package command lines

Virtual packages can be launched with the following arguments (case insensitive). Replace "AppName.cameyo.exe" with your virtual package's executable name:

Parameters:

- **-Stop: stop and remove virtual application (1.55 and higher)**

Terminates virtual application's currently-running processes.

Example: AppName.cameyo.exe -Stop

- **-Remove: stop and remove virtual application**

Stops and removes the virtual application and its local data from the machine.

Can be used with the "-Confirm" modifier, to request user confirmation.

To remove only the virtual registry, you can specify -Remove:Reg

Examples:

- AppName.cameyo.exe -Remove

- AppName.cameyo.exe -Confirm -Remove

- **-Exec: execute an application from the virtual package context**

Executes a command from the virtual application's package.

If no full path is provided (i.e. "skype.exe"), the entire virtual package tree will be searched.

A number such as "#1", "#2" can be specified to execute the first, second etc. item from the package's autolaunch menu.

Examples:

- AppName.cameyo.exe -exec "%Program Files%\OpenOffice.org 3\program\scalc.exe"

- AppName.cameyo.exe -exec scalc.exe Parameter1 Parameter2

- AppName.cameyo.exe -exec #1 (executes the first item from the package's autolaunch menu)

- AppName.cameyo.exe -exec regedit.exe (opens the registry editor in the context of the virtual package)

- AppName.cameyo.exe -exec cmd.exe (opens a command-line box in the context of the virtual package)

Old syntax (pre 1.6): AppName.cameyo.exe "-exec:%Program Files%\OpenOffice.org 3\program\scalc.exe"

- **-Integrate / -FullIntegrate: integrates applications registered associations & shortcuts (1.6 and higher)**

Integrates the application's registered extension associations. For example: double-click handler for .PDF files, in case of Adobe Reader.

Integrates the application's shortcuts into the real machine. For example: Start menu & Desktop shortcuts.

These items will be unregistered when the application is remove using the -Remove command.

-Integrate and -FullIntegrate are similar, except that -FullIntegrate first copies the virtual executable locally before performing the registration.

To perform integration for all users (rather than just the current user), -AllUsers can be specified before the -Integrate / -FullIntegrate parameters.

To undo integration, use -Unintegrate.

Examples:

- AppName.cameyo.exe -Integrate
- AppName.cameyo.exe -AllUsers -Integrate

- **-Troubleshoot: generates a troubleshooting log for virtual applications that fail to run**

The log generated by this parameter should be sent to cameyo @ cameyo.com to troubleshoot virtual applications that fail to run. When you send the log, please specify in the subject "Troubleshooting log". Also, please provide a direct link to the installer of this software.

- **-Jumpto: explore the application's virtual repository**

Opens a Windows Explorer window onto the application's virtual repository directory .

Advanced commands:

- **-CreateShortcuts: create a shortcut for each of the application's start menu**

If you wish to have standalone files that start the application with each of its start menu items, this parameter creates a link for each of the application's start menu entries. These links are created in the same directory as the virtual application's executable.

- **-Repackage: integrate the application's data storage into the virtual executable package (1.55 and higher)**

Takes all the deployed files and data from the local file system back into the virtual application package.

This is useful if you want the virtual package to always include the files and data as they are currently on the machine.

- **-Patch: update the virtual package with an installer / patch (1.6 and higher)**

Performs an installation / patch in the context of the virtual package, and repackages the result into AppName.virtual.new.exe.

Note: 32-bit packages can only be update on 32-bit machines.

Examples:

- AppName.cameyo.exe -Patch "C:\SomeDir\Patch.msi"
- AppName.cameyo.exe -Patch "C:\SomeDir\Patch.exe"
- AppName.cameyo.exe -Patch "C:\SomeDir\Patch.exe" Parameters

Expert commands:

- **-Break: breaks the virtual package into its different components (1.55 and higher)**

Breaks the different virtual package components into sub-directory "AppName.virtual.break"

- **-Unbreak: unbreaks the virtual package (1.55 and higher)**

The opposite of the -Break command. This command takes the components from the sub-directory "AppName.virtual.break" and puts them back together into AppName.cameyo.exe.

This command is for very advanced users; you should never have to modify the ZIP or DB components, as they are tied together. Inconsistency between the ZIP contents and the VirtFiles.db can cause the virtual package to misbehave.

- **-ChangeEngine: change Engine (1.6 and higher)**

Changes the virtual package's engine with another version. This is usually useful for updating an existing package with a newer Cameyo version's engine.

Example: AppName.cameyo.exe -ChangeEngine C:\NewCameyo\AppVirtDll.dll

Tip: You can achieve the same result by dragging & dropping the wanted AppVirtDll.dll over the Package Editor's window, while the package you want to update is being edited.

- **-ChangeLoader: change Loader (1.6 and higher)**

Changes the virtual package's loader with another version. This is usually useful for updating an existing package with a newer Cameyo version's loader.

Example: AppName.cameyo.exe -ChangeLoader C:\NewCameyo\Loader.exe

- **-EngineVer: display engine version (1.6 and higher)**

Example: AppName.cameyo.exe -EngineVer

- **-Info: display advanced package info, useful for troubleshooting (1.6 and higher)**

Example: AppName.cameyo.exe -Info

- **-ProcessList: display list of running virtual apps and processes (1.8 and higher)**

Example: AppName.cameyo.exe -ProcessList

- **-Properties: display list of running virtual apps and processes (2.5 and higher)**

Allows running virtual package with a modified set of properties. The syntax is:

AppName.cameyo.exe -Properties:Property1=Value1[,,Property2=Value2,,Property3=Value3]

The separator ",," allows specifying several properties, while authorizing single-comma to be specified within values.

Examples:

- AppName.cameyo.exe -Properties:BaseDirName=C:\Apps\MyApp

- AppName.cameyo.exe "-Properties:AppID=New application ID,,Expiration=31/12/2012"

- **-ExportProperties / -ExportAllProperties: exports package properties (2.6 and higher)**

Exports package's properties to an XML file.

-ExportProperties excludes properties which are specific the package: AppID, Build, BuildUID, CloudPkgId, EngineVersion, FriendlyName, Publisher, Version, Shortcuts. Hence it is useful for exporting properties from a package for re-importing them to other packages using Cameyo.exe -ImportProperties (see Packager command lines).

-ExportAllProperties saves all package properties, including build-specific identifiers. It is not recommended to later re-import these properties, as they include application-specific and package-specific items that should not be applied to other packages.

Example: AppName.cameyo.exe -ExportProperties:C:\Properties.xml

Note: you can define a default settings template for new packages you create, by placing such XML file in the same directory as Cameyo's Packager.exe and naming it DefaultProperties.xml.

Modifiers (used in combination with other parameters, must be specified first):

- **-Quiet: quiet mode, no output (1.6 and higher)**

Example: AppName.cameyo.exe -Quiet -Remove

- **-ErrorsOnly: errors display mode, no output except for errors (1.6 and higher)**

Return values

0 indicates success.

All other values indicate failure. See the SDK's APIRET_ return codes for more detail.

Environment variable: CAMEYO_BASEDIRNAME=C:\SomeDir

Can be used to change the default "VOS" storage location. This can be useful for example for specifying a local directory instead of the users profile in a domain setup.

Note: CAMEYO_BASEDIRNAME is only taken into account if the application's data storage location is set to (Default). Otherwise it is ignored.

Example (batch file):

set CAMEYO_BASEDIRNAME=%ExeDir%\settings1

AppName.cameyo.exe

3.2 Packager command lines

- **-ChangeEngine: replace a virtual package's virtualization engine.**

Replace a package's virtualization engine with a newer Cameyo engine without having to re-package it.

New Cameyo versions often bring improved virtualization. This command helps you take advantage of the new version without having to re-package the application.

Example: Cameyo.exe -ChangeEngine AppName.cameyo.exe "C:\Program Files\Cameyo\AppVirtDLL.dll"

Results in AppName.virtual.new.exe with the provided virtualization engine.

Tip: You can achieve the same result by dragging & dropping the wanted AppVirtDll.dll over the Package Editor's window, while the package you want to update is being edited.

- **-GhostCapture: captures an installation in "Ghost" mode.**

Captures an installation directly into the virtual sandbox, hence without truly installing anything on the host machine.

This mode is convenient for situations where you do not wish to install the software into the real machine. However, while it works for simple installations, more complex installations may fail.

Not supported on 64-bit machines.

Examples:

- Cameyo.exe -GhostCapture C:\SomeDir\Setup.msi

- Cameyo.exe -GhostCapture C:\SomeDir\Installer.exe

- Cameyo.exe -GhostCapture C:\SomeDir\Installer.exe Parameters

- **-SetProperties: sets / modifies virtual package properties**

You can modify a virtual package's properties from the command line. The syntax is:

Cameyo.exe -SetProperties:Property1=Value1[,,Property2=Value2,,Property3=Value3] ExePath

The separator ",," allows specifying several properties, while authorizing single-comma to be specified within values.

Examples:

- Cameyo.exe -SetProperties:AppID=NewAppID C:\AppName.cameyo.exe

- Cameyo.exe "-SetProperties:AppID=New application ID,,Expiration=31/12/2012" "C:\Some dir\AppName.cameyo.exe"

- **-ExportProperties: saves a set of properties from a virtual package to a template XML file (licensed versions)**

Exports package properties into an XML file.

Examples:

- Cameyo.exe -ExportProperties:C:\PropertiesTemplate.xml C:\AppName.cameyo.exe

- Cameyo.exe "-ExportProperties:C:\PropertiesTemplate.xml" "C:\Some dir\AppName.cameyo.exe"

- **-ImportProperties: applies a set of properties from a template XML file (licensed versions)**

Imports a package properties template from an XML file. Typically you can create the template XML file by using the -ExportProperties command on a package which you've configured as you like (using the Package Editor or the SetProperties command) and then apply it to other packages.

Examples:

- Cameyo.exe -ImportProperties:C:\PropertiesTemplate.xml C:\AppName.cameyo.exe

- Cameyo.exe "-ImportProperties:C:\PropertiesTemplate.xml" "C:\Some dir\AppName.cameyo.exe"

- **-StartCapture: saves a pre-snapshot of the system**

This parameter performs only the first half of the application packaging process. It creates a pre-snapshot, and saves it on disk.

When preceded by "-Quiet", runs in batch mode: no prompt and an exit code of 0 upon success.

Examples:

- Cameyo.exe -StartCapture

- Cameyo.exe -Quiet -StartCapture

- **-EndCapture: loads a previously saved snapshot, and produces a virtual application package**

This parameter loads the snapshot previously saved by -StartCapture, and produces a virtual app package out of it.

When preceded by "-Quiet", runs in batch mode: no prompt and an exit code of 0 upon success.

When preceded by "-OutputExe:full_file_name.exe", it forces the output executable name.

Examples:

- Cameyo.exe -EndCapture

- Cameyo.exe -Quiet -OutputExe:C:\SomeDir\App.cameyo.exe -EndCapture

- Cameyo.exe -Quiet "-OutputExe:C:\Some dir with spaces\App.cameyo.exe" -EndCapture

Packager configuration file (advanced)

The Packager's configuration file is Packager.xml, with default location: %AppData%\VOS\Cameyo\PROG\%Program Files%\Cameyo. It includes the following configuration items:

- OutputDir: path for saving the produced package.
- Compression: compression algorithm to use. Possible values include: none, snappy (default, optimized for speed), bzip2 (optimized for size).
- PathsToAvoid: file system paths to avoid during capture.
- KeysToAvoid: registry keys to avoid during capture.

Automating the packaging process using StartCapture & EndCapture:

The functions StartCapture and EndCapture allow automating the packaging process. Suppose you want to automate two applications:

- Execute & wait: Cameyo.exe -Quiet -StartCapture
- Install application #1
- Execute & wait: Cameyo.exe -Quiet -OutputExe:C:\App1.cameyo.exe -EndCapture
- Execute & wait: Cameyo.exe -Quiet -StartCapture
- Install application #2
- Execute & wait: Cameyo.exe -Quiet -OutputExe:C:\App2.cameyo.exe -EndCapture

3.3 Custom events (licensed versions)

You can configure Cameyo virtual packages to launch custom commands at different points during the application's execution.

The syntax for adding a script to a package is:

Cameyo.exe -AddScript ScriptFile EventName wait/nowait ExePath

With:

- ScriptFile: full path to the script file to add. If this is a vbs / wsh / js / jse file, it will be executed by the Windows Script host (wscript.exe). If it is a bat or cmd file, it will be executed using "cmd.exe /c". Otherwise it will be executed directly.
- EventName: the moment at which the script is to execute. See event tables below.
- wait / nowait: specify "wait" to make the application wait until the script is done executing. Specify "nowait" if you wish the application to proceed while the script executes in the background.
- ExePath: full path to the .cameyo.exe virtual package.

Examples:

- Cameyo.exe -AddScript C:\Scripts\DoSomething.vbs OnProcessStartUnvirtualized wait C:\AppName.cameyo.exe
- Cameyo.exe -AddScript C:\Scripts\DoSomething.vbs OnStartVirtualized nowait C:\AppName.cameyo.exe

Event name:	Triggered:	Order:
OnStartUnvirtualized	When the very first process of the application starts. The command	Loading #1

	executes outside the virtualization context.	
OnStartVirtualized	When the very first process of the application starts. The command executes inside the virtualization context.	Loading #3
OnProcessStartUnvirtualized	Whenever a new process starts. The command executes outside the virtualization context.	Loading #2
OnProcessStartVirtualized	Whenever a new process starts. The command executes inside the virtualization context.	Loading #4
OnProcessStopVirtualized	Whenever a process stops. The command executes inside the virtualization context.	Unloading #1
OnProcessStopUnvirtualized	Whenever a process stops. The command executes outside the virtualization context.	Unloading #4
OnStopVirtualized	When the very last process of the application ends. The command executes inside the virtualization context.	Unloading #2
OnStopUnvirtualized	When the very last process of the application ends. The command executes outside the virtualization context.	Unloading #3

The "order" column describes the order by which events are triggered during the executable process' loading / unloading sequence.

Virtualized / Unvirtualized

Each events can trigger before or after the virtualized environment is started / stopped. This allows you to choose whether to execute commands within or outside of the virtual application's context.

For example, if you need to run a command when a virtual application starts, but you need it to run outside of the sandbox (i.e. create files on the real system), then you can specify it in the "On start (unvirtualized)" event. If on the other hand you would like to execute a certain command when the virtual application ends, but you want it to still run inside the virtual environment, then you can specify it in the "On stop (virtualized)" event. The option "Wait until program ends" delays the virtual application's start or stop until your command has finished executing.

Information about the package properties

Regardless of which event triggered your script, your script will have detailed information about the package, its properties and its different paths, through Cameyo's standard Environment variables.

Stopping application execution

If your script runs in "wait" mode, it can issue the exit code 1234 to stop / cancel the application's execution.

3.4 Path variables

Cameyo uses variablized path naming. For example, the generic form of "C:\Program Files\abc.exe" is: "%Program Files%\abc.exe".

This convention makes paths work on any system regardless of Windows installation drive, language, customized profile and user paths etc.

Variable:	Example of target location:
%Program Files%	C:\Program Files
%System%	C:\Windows\System32

%Windows%	C:\Windows
%Fonts%	C:\Windows\Fonts
%Common Startup%	C:\Documents and Settings\All Users\Start Menu\Programs\Startup
%Common Programs%	C:\Documents and Settings\All Users\Start Menu\Programs
%Common StartMenu%	C:\Documents and Settings\All Users\Start Menu
%Common AppData%	C:\Documents and Settings\All Users\Application Data
%Common Desktop%	C:\Documents and Settings\All Users\Desktop
%Common Profile%	C:\Documents and Settings\All Users
%Local AppData%	C:\Documents and Settings\user\Local Settings\Application Data
%Startup%	C:\Documents and Settings\user\Start Menu\Programs\Startup
%Programs%	C:\Documents and Settings\user\Start Menu\Programs
%AppData%	C:\Documents and Settings\user\Application Data
%Desktop%	C:\Documents and Settings\user\Desktop
%Favorites%	C:\Documents and Settings\user\Favorites
%SendTo%	C:\Documents and Settings\user\SendTo
%Templates%	C:\Documents and Settings\user\Templates
%NetHood%	C:\Documents and Settings\user\NetHood
%Recent%	C:\Documents and Settings\user\My Recent Documents
%MyVideo%	C:\Documents and Settings\user\My Documents\My Videos
%MyPictures%	C:\Documents and Settings\user\My Documents\My Pictures
%Personal%	C:\Documents and Settings\user\My Documents
%Internet Temp%	C:\Documents and Settings\user\Temp\Temporary Internet Files
%Internet History%	C:\Documents and Settings\user\History
%Profile%	C:\Documents and Settings\user

System environment variables can be injected into a Cameyo path by using "%%". For example: "%%TEMP%%" is substituted by Windows' environment variable TEMP. So for example setting the virtual repository's location to "%%TEMP%%\Cameyo" is be translated into: "C:\Users\me\AppData\Local\Temp\Cameyo".

3.5 Virtual package properties

Cameyo virtual packages contain properties which determine their behavior. These properties are changed by the Package Editor. They can also be changed manually using the -SetProperties command line (see Packager command lines).

Here is a description of these properties:

General properties:

- AppID: virtual application ID. This is what defines the virtual application's "bubble". Two packages with different AppID can run side by side. Two packages with the same AppID are considered as an update

of one another, other when launched.

- VirtMode: virtualization mode. Either "RAM" or "Disk" (default is "RAM").
- IsolationMode: isolation mode. Can be "Data", "Isolated" or "FullAccess". Exceptions can be specified on the folder / file / registry key / registry value level through sandbox configuration.

Storage:

- BaseDirName: storage path. Default behavior is: if running from hard drive: "%AppData%\VOS\%AppID%". Otherwise, if running from removable drives or cloud drives such as DropBox, storage dir is: [executable name].data
- DataDirName: different path for the "CHANGES" sub-directory. Default: [BaseDirName]\CHANGES.

Startup menu:

- AutoLaunch: executable command that gets launched when the virtual package is executed. This property can encapsulate several auto-launch entries, separated by ";" in which case a launch menu will be displayed when the package is launched. Each auto-launch entry can contain additional (optional) information in the form: "Target>Arguments>Name>Description". For example, the following string will start a launch menu requesting the user to choose between Word and Excel: "%Program Files%\Office\Winword.exe>>Word;%Program Files%\Office\Excel.exe".
- AutoLaunchAlwaysAllowed: when this value is equal to "0", the "Always launch this program" checkbox will be hidden from the startup menu.

Advanced properties:

- StopInheritance: child processes to exclude from virtualization, separated by ";".
- StartingDir: starting directory.
- AutoUpdate: URL to auto-update server.
- EnvVariables: environment variables for the virtual application, in the syntax: "MYENV1=Something1". Several environment variables can be entered, separated by ":==:". For example: "MYENV1=Something1:==:MYENV2=Something2". "%ExistingValue%" can be used to refer to the environment variable's existing value, if any. For example: "PATH=%ExistingValue%;C:\AdditionalPath".
- VirtMachineName: machine name to simulate to the virtual application.
- VirtUserName: user name to simulate to the virtual application.
- RegMode: when set to "Extract", forces virtual registry to be extracted and redirected to HKCU\Software\VOS\[AppID]\Registry. When empty, virtual registry will not be stored on the HKCU registry hive at all (except on XP and during UAC process elevation).
- ExecWait: defines whether the loader (the .cameyo.exe process) should exit immediately after it starts the virtual app, or whether it should wait for the virtual process to finish and only then exit. This is useful for scripting, when you need to wait for the actual application process to finish its work. Value can be one of the following: "None": the loader quits immediately after running the virtual app process. "Always": loader waits for virtual app's process to finish. "CmdLine": loader waits only if the package runs a command line argument using the -exec parameter. "AutoLaunch": loader waits only if the package is used without a command line argument (without the -exec parameter).

Expiration properties:

- TtlDays: number of days before expiration.
- TtlResistRemove: "0" or "1" (default: "0"). Defines whether the above expiration mechanism should resist application removal + re-installation or not (=lost the expiration counter once the application is

removed).

- Expiration: expiration date in format: dd/mm/yyyy.

Custom events properties (please use Cameyo -AddScript command):

Custom events can be defined using the following syntax: command>parameters>[wait]. "wait" means that the application's process should wait for the script to finish execution. For more details, see Custom events (licensed versions). Several custom events can be entered for the same event, separated by ";". Custom event properties are named as following:

- OnStartVirtualized
- OnStartUnvirtualized
- OnStopVirtualized
- OnStopUnvirtualized
- OnProcessStartVirtualized
- OnProcessStartUnvirtualized
- OnProcessStopVirtualized
- OnProcessStopUnvirtualized

Troubleshooting properties:

- LegacyReg: "0" or "1" (default: "0"). Defines whether to force virtual registry to go into the registry's HKCU\Software\VOS\AppID\Registry key (the default behavior under XP and for UAC-elevated programs).
- EarlyInjection: "0" or "1" (default: "1"). Turns on or off an advanced mechanism that loads the virtualization engine very early during a process runtime, and which is relevant for Disk virtualization mode only.

Remarks

Most properties are optional, and leaving them empty keeps the default behavior.

3.6 Environment variables

Configurable variables:

Starting from Cameyo 2.0, you can configure any of Cameyo's INI settings from environment variables, using the "CAMEYO_" prefix.

For example, tu customize the "BaseDirName" INI setting (which controls the path of the virtual repository), you can set the environment variable CAMEYO_BASEDIRNAME. You can of course use Cameyo's Path variables.

As another example, if you would like to change the "AutoLaunch" INI setting (which controls the autolaunch / menu displayed when the virtual app is launched), modify the variable CAMEYO_AUTOLAUNCH.

The same goes with package expiration, custom events, as well as all other INI settings.

Example (batch file):

```
set CAMEYO_BASEDIRNAME=%%ExeDir%%\mypath
set "CAMEYO_AUTOLAUNCH=%%Program Files%%\Soft1.exe>>First menu item;%%Program Files%%\Soft2.exe>>Second menu item"
```

AppName.cameyo.exe

Note: the "%%" are .BAT file's syntax of representing a single percent ("%") character.

Read-only variables:

When virtual applications execute, Cameyo places certain environment information into their environment variables, prefixed with "CAMEYO_RO_" ("RO" as in "read only"). This is helpful for obtaining information about virtual apps from scripts and automation tools:

- CAMEYO_RO_VIRTUALAPP: this variable is always set to TRUE, to indicate that the application is currently running virtualized.
- CAMEYO_RO_APPID: the application's unique identifier (AppID)
- CAMEYO_RO_CARRIEREXE: the complete path of the virtual app's launcher executable.

Example values:

CAMEYO_RO_APPID=WordProcessor

CAMEYO_RO_CARRIEREXE=D:\MyApps\WordProc.cameyo.exe

CAMEYO_RO_VIRTUALAPP=TRUE

Troubleshooting

4 Troubleshooting

4.1 First troubleshooting steps

Virtual package failing to work properly

- Try running the application package with the "-Remove" parameter, and then run it with the "-SafeMode" parameter.
- If it doesn't help, see the Virtualization logs section for reporting your issue to the Cameyo team, if you are eligible for support. Please send along the package itself (i.e. via wetransfer.com) and if possible also the installer of the software itself (along with special installation instructions if needed).

4.2 Virtualization logs

Cameyo's support team may request that you submit a virtualization engine log. This helps diagnosing virtualization issues. There are two ways of generating these logs:

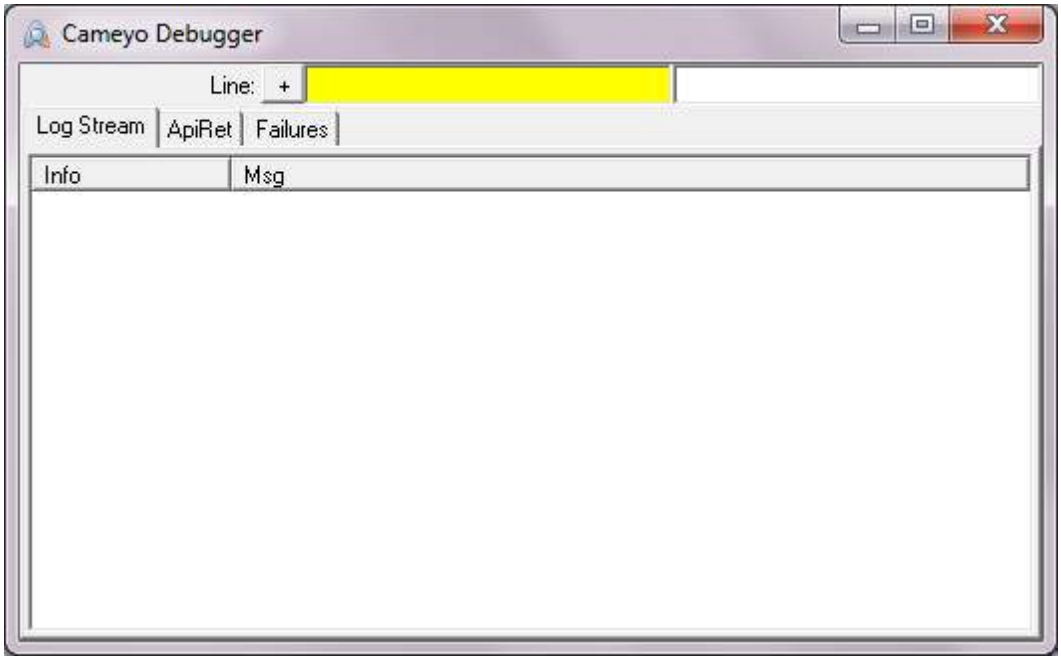
A. Virtualization log method #1 (-Troubleshoot)

By running the virtual application with the "-Troubleshoot" parameter, you will see the following screen. Simply reproduce the problem and click "Reproduced". Send the resulting log.

B. Virtualization logging method #2 (CameyoTracer)

This will generate a log similar to the -Troubleshoot method above. This procedure is helpful when the above doesn't work, or when you need to narrow the log to a specific point in time.

First you need to download and run www.cameyo.com/downloads/CameyoTracer.exe. The following screen will appear:



The procedure is then:

1. Launch the virtual application (CameyoTracer must be running before you start the virtual application).
2. Reproduce the problem. As soon as the problem is reproduced, quickly press Ctrl-E to stop logging. You will see the text "Logging" change into "Stopped logging".
3. Press Ctrl-S to save the log.
4. Compress and send the resulting log to the Cameyo team.

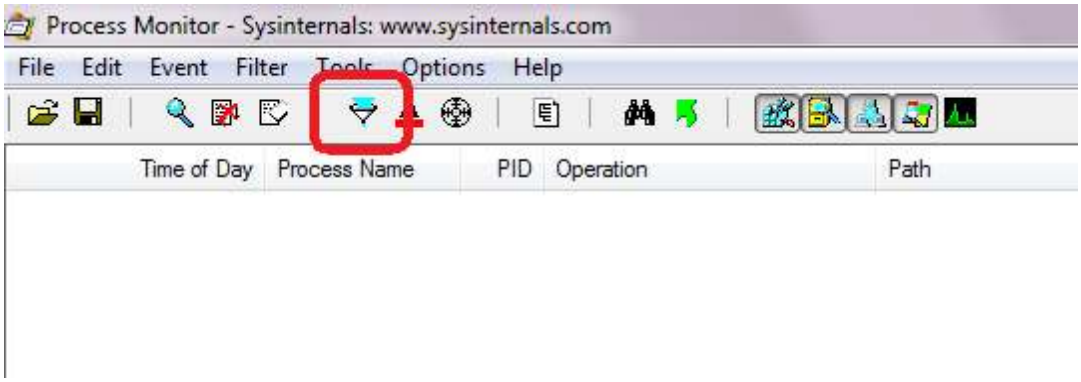
Important: CameyoTracer substantially slows down the virtual application's execution. If the application takes too long for you to get to the problematic event, you can disable logging (Ctrl-E) right before step 1, and re-enable it (Ctrl-E) just before step 2 (just before starting to reproduce the problem). This will yield a much shorter log and faster execution.

C. Process monitor

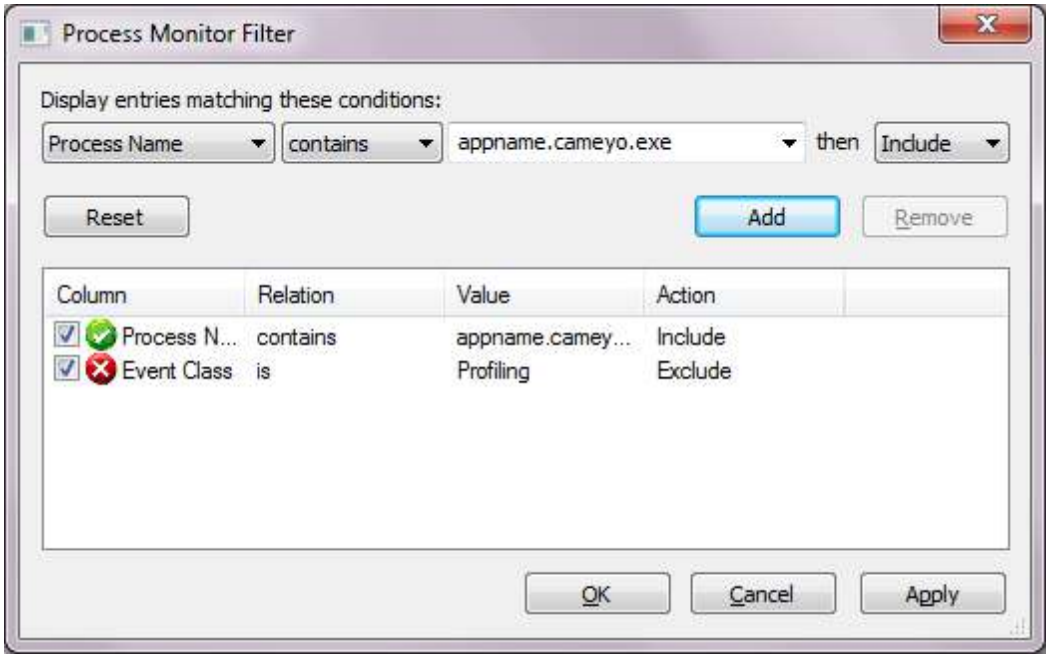
In addition to a virtualization log, it is often useful to provide a Process Monitor log to be able to thoroughly diagnose virtualization issues. Process Monitor is specifically helpful for diagnosing virtualization issues related to registry virtualization.

Process Monitor is a diagnosis tool from Microsoft TechNet. It can be downloaded from http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx

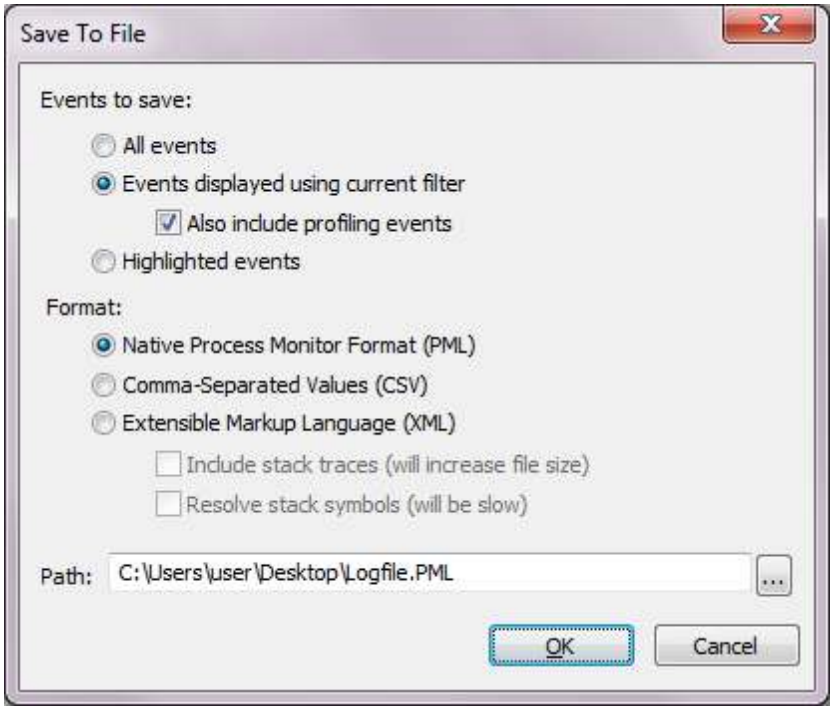
- Download and launch Process Monitor. Press the "Filter" button:



- The following filter screen will appear:

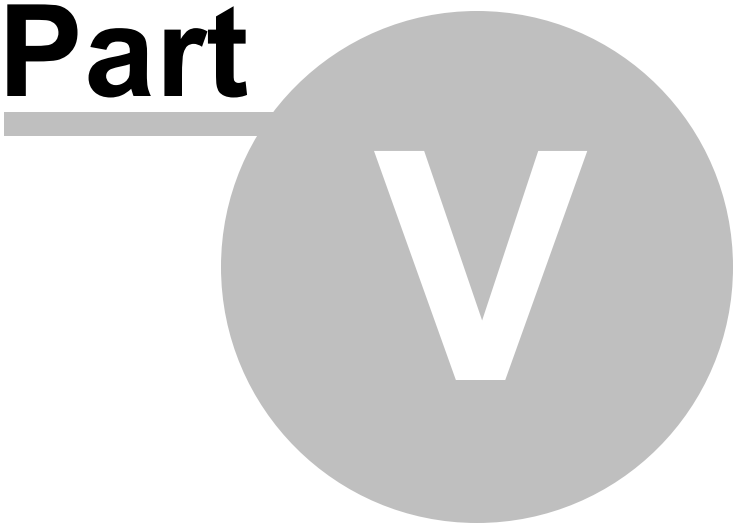


- Click the "Reset" button.
- Choose "Process Name" on the top left drop-down list. Then choose "contains" and then enter the relevant process name (see remark below). Click "Add", and "OK".
- Choosing the relevant process name is important: if your application is virtualized in RAM mode, this will generally be the same as the .cameyo.exe filename. If it is in Disk mode, it will be the same process name as the original (non-virtualized) application process. If you are unsure, try first to let Process Monitor run without any filter (Cancel and come back to it later), and see which of the virtual application's process produces the relevant log lines.
- Reproduce the problem.
- Stop the logging (Ctrl-E) and save the log file (Ctrl-S)



- Leave the saving dialog's options by default ("Events displayed using current filter" and "Native Process Monitor Format (PML)").
- Click "OK", compress resulting PML log file and send it to the Cameyo team.

Automated package builder (command-line)



5 Automated package builder (command-line)

5.1 Overview

Note: feature available to professional licenses only (Enterprise / Developer)

Cameyo's command-line builder (BuildPkg) is designed for software developers and publishers, allowing you to automate the packaging of virtual applications.

It can integrate with your software building process, producing a single executable your users can download and execute without prior installation. This packaged virtual executable will contain all of your application's ecosystem (files, DLLs, registry etc), just like standard Cameyo applications. The command-line builder was created using Cameyo's open-source SDK.

Remarks:

- Starting from version 2.5, the command-line builder replaces the module which was previously named "ZeroInstaller".
- Using the command-line builder requires an activated Cameyo license, or it will fail to function.

5.2 XML-based package building

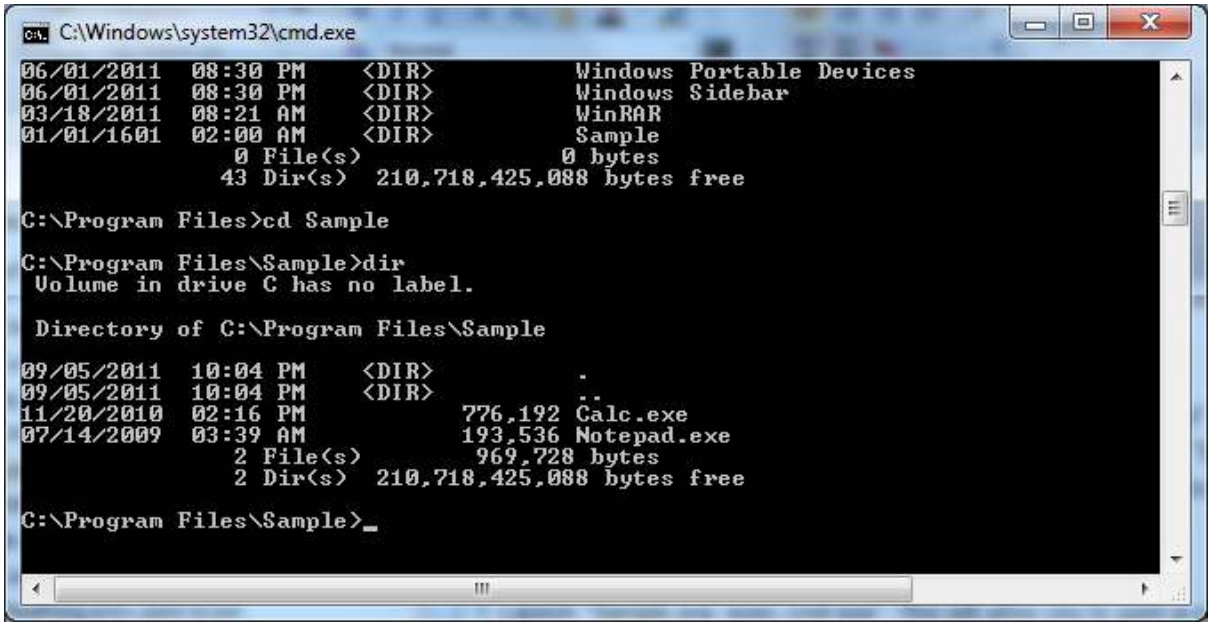
In this chapter you will find a hands-on, step-by-step tutorial for using Cameyo's command-line builder. In just a few minutes you will become familiar with the command-line builder and capable of packaging your own virtual applications.

Step 1: Build your first virtual application

1. If you haven't already, download Cameyo (i.e. to directory c:\cameyo) and activate its license. Activating the license requires that you launch the Cameyo menu and login, just once.
2. Download Cameyo's sample command line XML, i.e. to directory c:\cameyo. (If the link appears blank in your browser, be sure to show the page source).
3. From the command line, go to directory c:\cameyo by entering "cd \cameyo", and then run: "c:\cameyo\Cameyo.exe -BuildPkg BuildPkgSample.xml".
4. Congratulations! You have just created your first virtual application: "Sample.exe" (in the same directory as Cameyo.exe).
5. Now launch Sample.exe and you will see that you can start the Calculator or Notepad.

Step 2: Explore your virtual application

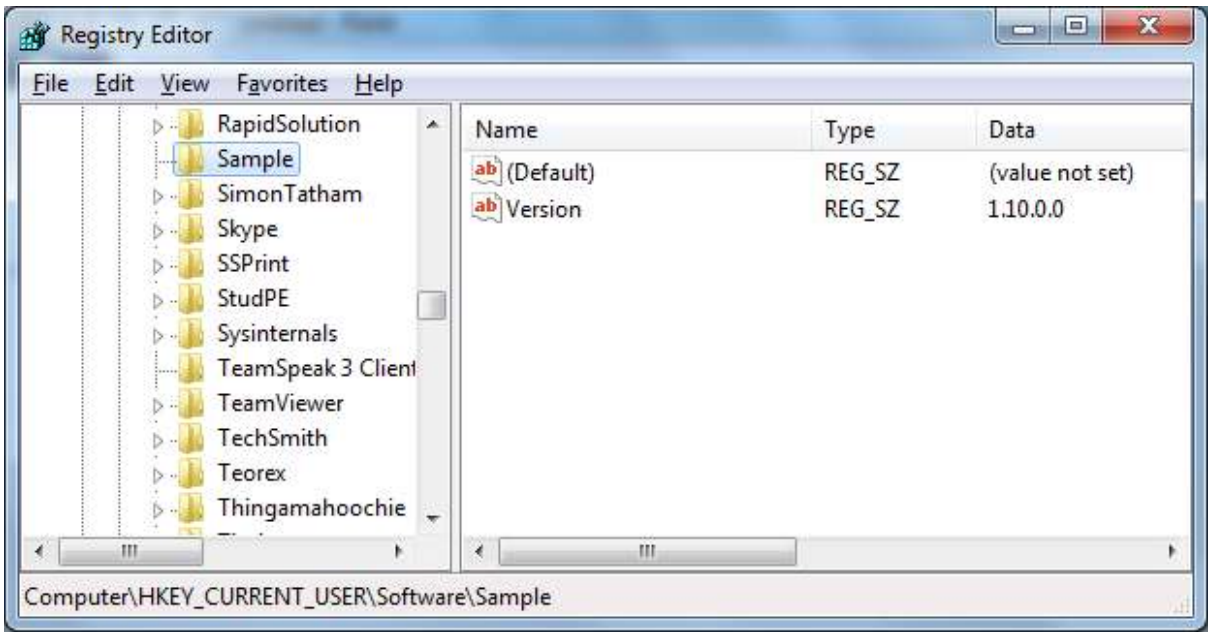
1. Launch "Sample.exe -exec cmd.exe". This will open a command line from the virtual application's context.
2. In the command line, go to your Program Files directory ("cd \program files") and list all files ("dir").
3. There you can see a directory called "Sample". That directory and its files do not really exist, they are virtual! Enter the Sample directory ("cd Sample") and "dir" again.
4. Now you can see the two executable files from BuildPkgSample.xml, as if they were really there. That's how file system virtualization will be perceived by the applications you package.



File-system virtualization: your application will see directories and files that don't really exist

Now let's do the same exploratory tour with the registry:

1. Launch "Sample.exe -exec regedit.exe" (make sure no instance of Regedit is already running). Again, this will show you the registry from the virtual application's angle:
2. Browse to HKEY_CURRENT_USER\Software\Sample. In there you will see an item called "Version" whose value is "1.10.0.0", just like defined in Sample.xml. Again, this value does not really exist there.
3. Close Regedit and open the real one instead. Now you will see that this key and its values do not really exist but were merely virtualized to the Sample.exe application:



Registry virtualization: your application sees things that do not truly exist

Step 3: Customize the application

1. Modify the IconFile and AppID properties, and rebuild. See how the changes affected your output executable.
2. Add FileSystem/File entries and Registry/Value entries, and rebuild. See the effects.
3. Change the Sandbox section as following, rebuild and see how your application is isolated differently:

```
<Sandbox>
        <FileSystem access="Isolated" />
        <FileSystem path="%Personal%" access="Full" />
        <Registry access="Full" />
        <Registry path="MACHINE\Software\Microsoft" access="Isolated" />
</Sandbox>
```

That's it!

You have now learned enough to get going with building your own virtual application. In the next chapter we will learn more about the different XML options.

Remember that you can use the BuildPkg command in any batch / scripting environment. Return code 0 means success, non-zero means failure.

Note: the actual process for the command line builder is named Packager.exe or Packager64.exe (Cameyo.exe simply executes it from the Cameyo's Program Files directory).

A more advanced example

Cameyo itself is packaged using the XML builder. Here is a portion of the XML which is used for building Cameyo:

```
<ZeroInstallerXml>
        <Properties>
                <Property AppID="Cameyo" />
                <Property FriendlyName="Cameyo" />
                <Property Version="2.1.968" />
                <Property IconFile="CameyoMenu.exe" />
                <Property StopInheritance="Packager.exe;Packager64.exe" />
                <Property VirtMode="DISK" />
                <Property StartingDir="Original" />
                <Property BuildOutput="[AppID].exe" />
                <Property CloudPkgId="1" />
                <Property AutoUpdate="https://online.cameyo.com/update.aspx" />
        </Properties>
        <FileSystem>
                <File source="AppVirtDll.dll" targetdir="%Program Files%\Cameyo" />
                <File source="AppVirtDll64.dll" targetdir="%Program Files%\Cameyo" />
                <File source="Loader.exe" targetdir="%Program Files%\Cameyo" />
                <File source="Packager.exe" targetdir="%Program Files%\Cameyo"
autoLaunch="Capture an installation" />
                <File source="..\..\cameyo-opensrc\Release\PackageEditor.exe" targetdir="%Program
```

```
Files%\Cameyo" />
            <File source="..\..\cameyo-opensrc\Release\PackageEditor.exe.config" targetdir="%
Program Files%\Cameyo" />
            <File source="..\..\cameyo-opensrc\Release\CameyoSDK.NET.dll" targetdir="%
Program Files%\Cameyo" />
            <File source="..\..\cameyo-opensrc\Release\Cameyo.OpenSrc.Common.dll"
targetdir="%Program Files%\Cameyo" />
            <File source="CameyoMenu.exe" targetdir="%Program Files%\Cameyo" />
            <File source="CameyoMenu.exe.config" targetdir="%Program Files%\Cameyo" />
        </FileSystem>
        <Registry>
            <Key path="%CurrentUser%\Software\Cameyo">
                <Value name="Version" string="[Version]" />
            </Key>
        </Registry>
        <Sandbox>
            <FileSystem access="Full" />
            <Registry access="Full" />
        </Sandbox>
</ZeroInstallerXml>
```

5.3 XML reference

Properties section:

Example:

```
<Properties>
      <Property AppID="Cameyo" />
      <Property Version="1.60.0.0" />
      <Property IconFile="Packager.exe" />            <!-- Copy the package icon from
      <Property ExpandSource="TRUE" />                <!-- i.e convert source="%Windo
```

This section defines properties in the form: <Property PropertyName="Value" />

There are 3 types of properties you can define here, depending on the property's name:

1. Special build properties (see below), some of which are required.
2. Override Cameyo's special properties to take effect during the application's execution. See Cameyo's properties.
3. Define any arbitrary properties you can use throughout the XML itself. For example, if you define a property called "MyOwn=ABC.TXT", then later on you can reuse that property later in the XML file, i.e: <File source="C:\[MyOwn]" />, or even in the definition of another property.

Special ZeroInstaller properties:

- AppID: application's name. Defines both the output executable's name and the virtual package's AppID property (Cameyo's internal application ID). The name must only contain legal filename characters.
- IconFile: file from which the output executable's icon is to be copied.

- BuildOutput (optional): name of the output executable. If no name is defined, the default name is AppID property + .exe.
- InputDir (optional): working path for the FileSystem section.
- ExpandSource (optional, default: false): indicates whether you'd like to expand XML file source names. For example, if you'd like <File source="%Windows%\Notepad.exe" /> to expand into C:\Windows\Notepad.exe, then you'd specify "true". If you'd like to keep the directory named "%Windows%\Notepad.exe" (for example if that is a sub-directory of your project dir), then you can leave it on "false" (default behavior).

FileSystem section:

Example:

```
<FileSystem>
      <File source="Loader.exe" targetdir="%Program Files%\Cameyo" />
      <File source="Packager.exe" targetdir="%Program Files%\Cameyo" />
      <File source="Launcher.exe" targetdir="%Program Files%\Cameyo" autoLaunch="[Appl
</FileSystem>
```

This section defines virtual files that are to be part of the application's package. They will be shown (virtualized) to the target application as if they exist in the indicated path.

Syntax: <File source="..." target/targetdir="..." />, where:

source: path to the file to include in the package. The path is relative to the working directory. If the property InputDir was defined, then this path is relative to it. Otherwise, it is relative to the current directory (path where ZeroInstaller.exe was launched from).

- target / targetdir: defines where the application needs the file to appear (virtually) to the application. "target" specifies the target directory + file name. "targetdir" only defines the target directory (keeping the filename portion same as the source).
- autoLaunch (optional): if defined, this file will be part of the application's auto-launch list. If the auto-launch list contains only one executable, it will be executed when the application is launched. If it contains more executables, a startup menu will be displayed for the user to choose which program to launch. The value of the autoLaunch property sets the text displayed to the user when the menu is displayed.
- autoLaunchArgs (optional): provides execution parameters for autoLaunch.

Paths can include Cameyo's Path variables.

Registry section:

You can define registry values in one of two ways. Either by specifying a full path in the "Value" node:

```
<Registry>
      <Value path="%CurrentUser%\Software\Cameyo\Version" string="[Version]" />
      <Value path="MACHINE\Software\Cameyo\SomeValue" string="Hello world" />
</Registry>
```

Or by embedding the Value node within a Key node:

```
<Registry>
      <Key path="%CurrentUser%\Software\Cameyo">
            <Value name="Version" string="[Version]" />
      </Key>
</Registry>
```

This section defines virtual registry keys that are to be part of the application's package. They will be shown (virtualized) to the target application as if they exist in the indicated path.

Syntax: <Value path="..." string="..." /> or: <Key path="..."><Value name="..." string="..." /></Key>

Supported value types:

- string="abc"
- dword="12af" equals hexadecimal number 0x12af
- bin="1234abcd" equals binary hex data: 12 34 ab cd

Note that registry keys are indicated in native paths.

Sandbox section:

Example:

```
<Sandbox>
      <FileSystem access="Isolated" />                            <- Default sandboxi
      <FileSystem path="%AppData%" access="Full" />               <- Full: read/write
      <FileSystem path="%Personal%" access="StrictlyIsolated" />  <- StrictlyIsolated
      <Registry access="Full" />
      <Registry path="MACHINE\Software\Microsoft" access="Isolated" />
</Sandbox>
```

This section defines how the virtual application interacts with the host system's file-system and registry: full access, or isolated access modes.

Syntax: <FileSystem/Registry path="..." access="..." />

Beware however that if a folder already exists in your virtual application, then any changes will go there. For example if your virtual application has a folder named %AppData%\MyApp, then any files saved by the application into %AppData%\MyApp will go into the virtual app's sandbox, not into the real machine, regardless of the sandbox isolation mode.

Note that registry keys are indicated in native paths.

FAQ

Part VI

6 FAQ

What is the recommended way for packaging software?

The easiest way to package software is by using our Online Packager directly from the cloud. It is ideal because the packaging environment is clean and optimized for application capturing. However not all software can be captured this way. So the second best method is to use a clean virtual machine (preferably XP SP3 32-bit), and use Cameyo's packager on it ("Capture an installation").

Cameyo is probably the product that provides most different methods for packaging virtual applications:

- Online Packager: usually the quickest way if your installation can be installed this way, and assuming it doesn't depend on pre-requisite components.
- VM packaging (Cameyo's local packager): if you have a clean virtual machine environment, that is usually the most recommended way.
- XML / command-line-based packaging: if you need to build packages automatically (i.e. via scripts).
- GhostCapture: the least recommended way, this method is about running the application's installer inside Cameyo's sandbox, and then wrapping the result into a virtual package. This method is useful because it can quickly be used on your own machine, however it is the least recommended way as it will often fail and / or produce less portable packages due to the unclean capture environment.

I do not see the files saved by Cameyo virtual applications.

Cameyo packages run with certain isolation configuration. By default (Data mode), only files saved under the Documents or Desktop folders, or on network / removable drives, will indeed be written there. All other files will be redirected into the application's isolation repository (the VOS directory).

My virtual application doesn't work properly. What should I do?

First, try running your app with the "-SafeMode" parameter. See the procedure described here: http://www.cameyo.com/resources/IssueSubmission.pdf

Can packages that require .NET framework run on machines that do not have the .NET framework?

Packages requiring the .NET framework must have at least some version of the .NET framework on target systems in order to work.

A virtual application does not work well. What should I do?

First, make sure you have the latest Cameyo version. If so, submit your application name on the forum, under the topic "Virtual applications not working well".

I've built my package, but I forgot to make some configuration changes to my software. How can I change settings and apply them to the package?

You can always go back to the packaging machine where the installer was capture, make some changes to your software, and redo a post-snapshot using the -EndCapture command (Cameyo 2.6.1182 and higher only).

How to specify path names in a package in a generic way?

Using path variables. Cameyo has its own path variables. For example, %System% expands to

something like C:\Windows\System32 (or D:\ or wherever Windows is installed on the target machine). You can also use Windows standard system variables by using double-percent sign: %% LOGONSERVER%%. For more examples, see the Path variables section.

How is Cameyo different from other products?

The main differentiators of Cameyo are:

- "Agent-less" virtualization, meaning virtual applications can run on machines that don't have any Cameyo agent pre-installed.
- Very simple to learn and use, yet very powerful if you need "under the hood" functionality.
- Provides many packaging methods.
- Automation and customization: almost anything can be done in Cameyo via command-line or scripts.
- Self-contained: you can customize and edit your packages without having to maintain heavy "project files / directories".
- We like to think of our active community as a strong point as well.

How to set up a Firewall exception rule for virtual applications?

Some firewalls, including the Windows Firewall, require to switch your virtual package to work in Disk mode. You can then see which executable causes the alert (within the VOS\... path) and add its full path as an exception. Beware: the Windows Firewall's group policy editor does not work well with paths containing percent signs (%). So for example if your executable is beneath ...\VOS\PROG\%Program Files%, the rule may not work. In this case it is best to install your virtual app in a custom path (so that it appears in the virtual package as C:\MyApp rather than C:\Program Files\MyApp). This will translate to a physical path such as: ...\VOS\PROG\C_ hence avoiding the percent (%) sign.

COMMERCIAL / LICENSING QUESTIONS

Is Cameyo open-source?

Only specific parts of Cameyo are open-source: the Package Editor and the SDK headers. But Cameyo is not open-source.

Is Cameyo free?

Cameyo's licensing is counted per number of end-users using the virtual apps that you create. It's free for less than 50. Above that number see our pricing page.

What is the difference between the Enterprise edition and the Developer edition?

The difference is mostly in terms of licensing and pricing model. An Enterprise license is for a specific number of end-users, without limitation as to the number of virtual applications. A Developer license is for a specific number of products (typically one), for a very large amounts of end-users.

I'm not a company or don't have 50 machines. How can I obtain the extended, licensed version?

The extended version can only be purchased by developers or companies with 50 computers or more. If this isn't the case for you, then the only way you can obtain the professional version of Cameyo is by contributing. There are many ways you can contribute: being an active tester, helping other users on the forum on a regular basis, translating to a new language, writing descriptions for our virtual app library,

developing useful add-ons, or other ideas you may have for contributing to the whole Cameyo community.

Acknowledgement

7 Acknowledgement

Version 2.5 was made with help from:

- **Mule** for his testings, helping users, dedication and commitment.

- **Max (Black751)** for helping with communication and localization coordination.

- Translations: Chinese: Shannon & Chenall, Indonesian: AdeKuching, French & Spanish: Max (Black751), German: Armin Wiesmüller, Lintux.

Uses the following components and modules:

- XmlParser, (c) 2002, Business-Insight.
- PNG Delphi, (c) Gustavo Huffenbacher Daud.
- SystemTray, (c) Chris Maunder.
- Info-Zip, (c) Info-Zip: license here.

License

Part VII

8 License

This is the general-purpose license for Cameyo. For developers / software publishers, please refer to this different license agreement: http://www.cameyo.com/license_dev.aspx

---

This Cameyo(CYO) End-User License Agreement ("EULA") is a legal agreement between you ("you") and CYO for the software product ("Software") mentioned above. By installing, copying, or otherwise using the Software, you agree to be bound by the terms of this EULA. The Software is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The Software is licensed, not sold, and ALL SOFTWARE IS PROVIDED "AS IS."

1. GRANT OF LICENSE: This EULA grants you the following rights:

· Installation and Use: You may install and use an unlimited number of copies of the Software. Any copies of the Software which this License authorizes you to make are subject to this EULA.

· Reproduction and Distribution: You may freely reproduce and distribute copies of the Software on up to 50 computers or user sessions (whichever is less); beyond that, you will need to acquire a license. Reselling the Software or integrating it into another software requires written consent by CYO. The CYO opening splash screens should never be tampered or removed.

2. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS:

· Limitations on Reverse Engineering, Decompilation, and Disassembly: You may not reverse engineer, decompile or disassemble the Software except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

· Support Services: CYO will not provide you with support services ("Support Services") related to the Software.

· Software Transfer: You may permanently transfer all of your rights under this EULA, provided the recipient agrees to the terms of this EULA.

· Termination: Without prejudice to any other rights of CYO's, your rights under this EULA will terminate automatically without notice from CYO if you fail to comply with any term(s) of this EULA. In such event you must destroy all copies of the Software and any related materials.

. Packaged software: Software packaged using CYO is free of royalty, not owned by CYO, and is under your sole responsibility. CYO is merely a packaging tool, and is in no way responsible of the use you make with packaged software. It is your responsibility to comply with copyright restrictions.

3. COPYRIGHT: All right, title and interest in and to the Software (including the copyrights in the Software and including, but not limited to, any images, photographs, animations, video, audio, music, text, and "applets" incorporated into the Software), the accompanying printed materials (if any), and any copies of the Software are owned by CYO. The Software is protected by copyright laws and international treaty provisions. Except as permitted by applicable law and this License, you may not modify, rent,

lease, loan, sublicense, or create derivative works from the Software.

4. Disclaimer of Warranty on Software. You expressly acknowledge and agree that use of the Software is at your sole risk. The Software is provided "AS IS" and without warranty of any kind and CYO EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. CYO DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE SOFTWARE WILL MEET YOUR REQUIREMENTS, OR THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT DEFECTS IN THE SOFTWARE WILL BE CORRECTED. FURTHERMORE, CYO DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE SOFTWARE OR RELATED DOCUMENTATION IN TERMS OF THEIR CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY CYO OR AN AUTHORIZED REPRESENTATIVE OF CYO'S SHALL CREATE A WARRANTY OR IN ANY WAY INCREASE THE SCOPE OF THIS WARRANTY. SHOULD THE SOFTWARE PROVE DEFECTIVE, YOU (AND NOT CYO OR AN AUTHORIZED REPRESENTATIVE OF CYO'S) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THE ENTIRE RISK ARISING OUT OF USE OR PERFORMANCE OF THE SOFTWARE REMAINS WITH YOU.

5. Limitation of Liability. UNDER NO CIRCUMSTANCES, INCLUDING NEGLIGENCE, SHALL CYO OR ITS SUPPLIERS BE LIABLE FOR ANY INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF OR RELATING TO THIS LICENSE, EVEN IF CYO HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS DO NOT ALLOW THE LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THIS LIMITATION MAY NOT APPLY TO YOU. IN NO EVENT SHALL CYO'S TOTAL LIABILITY TO YOU FOR ALL DAMAGES EXCEED THE AMOUNT PAID FOR THIS LICENSE TO THE SOFTWARE. THE ENTIRE RISK ARISING OUT OF USE OR PERFORMANCE OF THE SOFTWARE REMAINS WITH YOU.

6. Fault Tolerance. The Software is not fault tolerant and is not designed, manufactured, or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as the operation of nuclear facilities, aircraft navigation or communications systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the Software could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). CYO specifically disclaims any express or implied warranty of fitness for High Risk Activities.

7. Export Law Assurances. You may not use or otherwise export or reexport the Software except as authorized by United States law and the laws of the jurisdiction in which the Software was obtained. In particular, but without limitation, none of the Software may be used or otherwise exported or reexported (i) into (or to a national or resident of) a United States embargoed country or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nationals or the U.S. Department of Commerce's Table of Denial Orders. By using the Software, you represent and warrant that you are not located in, under control of, or a national or resident of any such country or on any such list.

8. Government End Users. If the Software is supplied to the United States Government, the Software is classified as "restricted computer software" as defined in clause 52.227-19 of the FAR. The United States Government's rights to the Software are as provided in clause 52.227-19 of the FAR.

9. MISCELLANEOUS: If for any reason a court of competent jurisdiction finds any provision, or portion of this EULA, to be unenforceable, the remainder of this EULA shall continue in full force and effect. This EULA constitutes the entire agreement between the parties with respect to the use of the Software and supersedes all prior or contemporaneous understandings regarding such subject matter. No amendment to or modification of this EULA will be binding unless in writing and signed by CYO.

3rd party libraries

Part IX

9 3rd party libraries

Cameyo is using the following 3rd party components:

Info-Zip license:

This program includes Info-Zip Software which was used by Cameyo pursuant to the following license.

This is version 2000-Apr-09 of the Info-ZIP copyright and license. The definitive version of this document should be available at ftp://ftp.info-zip.org/pub/infozip/license.html indefinitely.

Copyright (c) 1990-2000 Info-ZIP. All rights reserved.

For the purposes of this copyright and license, "Info-ZIP" is defined as the following set of individuals:

Mark Adler, John Bush, Karl Davis, Harald Denker, Jean-Michel Dubois, Jean-loup Gailly, Hunter Goatley, Ian Gorman, Chris Herborth, Dirk Haase,Greg Hartwig, Robert Heath, Jonathan Hudson, Paul Kienitz, David Kirschbaum,Johnny Lee, Onno van der Linden, Igor Mandrichenko, Steve P. Miller,Sergio Monesi, Keith Owens, George Petrov, Greg Roelofs, Kai Uwe Rommel,Steve Salisbury, Dave Smith, Christian Spieler, Antoine Verheijen,Paul von Behren, Rich Wales, Mike White

This software is provided "as is," without warranty of any kind, express or implied. In no event shall Info-ZIP or its contributors be held liable for any direct, indirect, incidental, special or consequential damages arising out of the use of or inability to use this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

Redistributions of source code must retain the above copyright notice, definition, disclaimer, and this list of conditions.

Redistributions in binary form must reproduce the above copyright notice, definition, disclaimer, and this list of conditions in documentation and/or other materials provided with the distribution.

Altered versions--including, but not limited to, ports to new operating systems, existing ports with new graphical interfaces, and dynamic, shared, or static library versions--must be plainly marked as such and must not be misrepresented as being the original source. Such altered versions also must not be misrepresented as being Info-ZIP releases--including, but not limited to, labeling of the alteredversions with the names "Info-ZIP" (or any variation thereof, including, but not limited to, different capitalizations), "Pocket UnZip," "WiZ" or "MacZip" without the explicit permission of Info-ZIP. Such altered versions are further prohibited from misrepresentative use of the Zip-Bugs or Info-ZIP e-mail addresses or of the Info-ZIP URL(s).

Info-ZIP retains the right to use the names "Info-ZIP," "Zip," "UnZip," "WiZ," "Pocket UnZip," "Pocket Zip," and "MacZip" for its own source and binary releases.

AjaxControlToolkit license:

This program includes AjaxControlToolkit Software which was used by Cameyo pursuant to the following license.

Copyright (c) 2009, CodePlex Foundation

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of CodePlex Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Google Snappy compression algorithm license:

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.