



Cloudpaging Studio 9.4 Guide

Proprietary and confidential to Numecent Inc.

This document cannot be reproduced in any fashion without advanced written consent of the company.

COPYRIGHT NOTICE

Copyright © 1996 - 2024 Numecent Inc.
530 Technology Drive, Suite #375, Irvine, California 92618 USA

All rights reserved.

This document is protected by copyright and contains information proprietary to Numecent Inc. Any copying, adaptation, distribution, or public display of this document without the express written consent of Numecent Inc. is strictly prohibited.

TRADEMARK NOTICE

Numecent, the Numecent logo, Cloudpaging, the Cloudpaging logo, and Cloudpager are trademarks or registered trademarks of Numecent Inc.

Microsoft, Windows, Microsoft Office, Microsoft Access, and Visual Studio are registered trademarks of Microsoft Corporation. All other names are used for identification purposes only and are trademarks or registered trademarks of their respective companies.

PATENT NOTICE

Protected by U.S. Patents: 6,453,334; 6,918,113; 6,959,320; 7,043,524; 7,062,567; 7,096,253; 7,149,761; 7,240,162; 7,451,196; 7,577,751; 8,024,523; 8,261,345; 8,359,591; 8,438,298; 8,509,230; 8,527,706; 8,661,197; 8,676,938; 8,752,128; 8,782,778; 8,831,995; 8,874,699; 8,892,738; 8,893,249; 8,898,391; 8,949,820; 9,054,962; 9,054,963; 9,072,972; 9,083,774; 9,094,480; 9,130,953; 9,300,752; 9,358,460; 9,380,063; 9,386,057; 9,436,578; 9,485,304; 9,497,280; 9,517,410; 9,571,501; 9,578,075; 9,654,548; 9,661,048; 9,675,890; 9,699,194; 9,716,609; 9,781,007; 9,825,957; 9,826,014; 9,838,449; 10,009,399; 10,021,168; 10,057,268; 10,356,100; 10,445,210; 10,587,473; 10,635,715; 11,119,884; 11,121,928; 11,451,548; 11,586,672; 11,740,992; 11,824,723

Patents pending in the U.S. and other countries.

Product version: 9.4.0

Guide revision: 1

Last update: 15 February 2024

NOTE: The product version syntax for Cloudpaging Web Portal Integration is: **M.m.p.**

(M: major version number, m: minor version number, p: patch version number)

Table of Contents

COPYRIGHT NOTICE	1
Table of Contents	2
Introduction	9
Features of Cloudpaging Studio	10
Guide Conventions	10
Getting Support	11
Website	11
Email	11
Chapter 1: Cloudifying Overview	12
Cloudifying Environment	13
Target Audience	13
Recommended Skills and Knowledge	13
Data Needed to Cloudify an Application	13
Application Assets	14
Configuration Information	14
The Cloudifying Project	14
Names for a Cloudified Application	14
Core Concepts	15
Virtualization	15
Prefetching	15
Compression	16
Encryption	16
The Cloudpaging Studio UI	16
The Main Window	16
Workspaces and Views	17
The Project Workspace	19
The AIB Workspace	19
The Token Workspace	19
Cloudifying Phases	21
Phase 1: Prepare the Environment	21
Phase 2: Capture the Application Installation	21
Phase 3: Cloudify the Application	22
Phase 4: Test and Optimize the Appset	22

Repackaging, Modifying, or Patching Appsets	23
How To Deploy Updated Appsets	24
Reusing Project Files	24
Appset File Structure.....	24
Chapter 2: Preparing the Environment.....	26
Cloudifying PC Technical Requirements.....	27
Target PC Technical Requirements	27
About Cloudifying Performance	27
Preparing a Clean Cloudifying PC.....	28
Installing Cloudpaging Studio	29
Chapter 3: Capturing the Application Installation	30
Procedure for Capturing the Application Installation.....	31
Step 1: Create a new Project.....	31
Step 2: Enter Basic Application Information	31
Step 3: Start the Application Installation Capture Process	35
Step 4: Install the Application	37
Step 5: Run the application (optional)	39
Step 6: Capturing System Reboots.....	39
Step 7: Stop the Installation Capture.....	39
Step 8: Enter Remaining Application Information.....	41
Step 9: Save the Project.....	43
About OS Compatibility Settings	43
Chapter 4: Cloudifying the Application	46
Procedure for Cloudifying the Application.....	47
Step 1: Configure Cloudpaging Settings.....	47
Step 2: Modify Files, Registry, and Other Assets	47
Step 3: Cloudify the Application	48
Chapter 5: Testing and Optimizing the Appset.....	50
Testing the Appset	51
Step 1: Test the appset.....	51
Step 2: Identify and Solve Problems	52
Optimizing the Appset.....	52
Step 1: Create Prefetch Files.....	52
Step 2: Set Compression	55
Step 3: Set Encryption.....	55

Step 4: Cloudify the Application	55
Step 5: Optional: Save the Cloudifying Environment.....	55
Step 6: Publish the Appset to Cloudpaging Server.....	56
Chapter 6: Cloudpaging Settings	57
Configurable AppEvents Settings	58
To add an AppEvent.....	58
About AppEvent Scripts and Executables	61
About AppEvent Context.....	62
Virtualization Settings	62
Sandboxing Settings	64
To add a folder exclusion	65
To add a registry key exclusion	66
Security Settings	66
To set up a security override:	66
Chapter 7: Modifying Project Assets	69
Modifying Files and Folders	70
About Folders	70
Adding Root Folders	72
Adding Sub-folders	73
Renaming Folders	75
Editing Folder Properties.....	75
Setting the Working Folder	76
Making Root Folders Merged.....	77
Adding Files.....	77
Editing Files.....	77
Editing File Properties.....	78
Deleting Files and Folders	79
Excluding Files and Folders.....	79
Setting the Command Line	79
Locating Files/Folders in Windows Explorer	79
Modifying Registry Entries.....	80
Adding Registry Keys	80
Renaming Registry Keys	82
Changing the Disposition of a Registry Key	83
Adding Registry Values	83

Editing Registry Values	84
Deleting Registry Keys and Values	84
Excluding Registry Keys	84
Modifying Environment Variables	85
Adding Environment Variables	86
Editing Environment Variables	86
Deleting Environment Variables.....	87
Excluding Environment Variables.....	87
Modifying Fonts	87
Adding Fonts.....	88
Editing Fonts	89
Editing the File Properties of a Font.....	90
Deleting Fonts.....	90
Excluding Fonts	91
Locating the File or Registry Entry of a Font	91
Modifying Startup Items.....	91
Types of Startup Items	92
Adding Startup Items.....	93
Editing Startup Items	94
Command Line File Properties of a Start Menu Item	94
Deleting Startup Items.....	95
Excluding Startup Items	95
Locating the File or Registry Entry of a Startup Item	95
Modifying Services and Drivers	96
Adding Services	97
Editing the Registry Key Properties of a Service.....	97
Editing the Image File Properties of a Service	97
Editing Services.....	98
Setting Virtualization Actions for a Service	99
Changing the Failure Action of a Service.....	100
Deleting Services	100
Excluding Services.....	100
Locating the Image File or Registry Key of a Service.....	101
Modifying Shortcuts.....	101
Adding Shortcuts.....	102

Editing Shortcuts	102
Setting the Command Line	103
Deleting Shortcuts.....	104
Excluding Shortcuts	104
Locating the source or target for a Shortcut	104
Searching for Assets.....	105
Finding Assets.....	105
The Find Results Window	107
About Search Types	108
Globally Changing Asset Dispositions.....	108
To Globally Change Dispositions:.....	109
Understanding Color Coding in Views	110
Conflict Color Coding.....	111
Chapter 8: Modifying an Appset.....	113
Opening an Appset file	114
To open an appset's Token or AIB file:	114
Editing an AIB file.....	114
Modifying File Settings.....	114
Exporting File Information	116
Searching for File Information	116
Modifying Registry Settings.....	117
Exporting Registry Information	118
Finding Registry Information	118
Modifying Minimum Cache Allocation.....	118
Editing Virtualization Information	119
Adding Prefetch Files	120
Viewing Prefetch Information.....	121
To view the prefetch statistics:	121
Editing Compatibility Settings.....	121
Editing Token files	122
Modifying an Application Value.....	122
Modifying OS Compatibility Settings	123
Modifying a Paging Source.....	124
Modifying a Configurable AppEvent	126
Modifying Sandboxing Settings.....	129

Modifying Virtualization Settings Flags.....	130
Modifying an Environment Setting	131
Modifying Security Override Settings	133
Saving an Appset File	134
To save an appset file using the 7-Zip Command Line.....	134
To save the appset using the 7-Zip GUI.....	134
Chapter 9: Patching an Appset	136
Patching an Appset	137
To patch an existing appset:	137
To add a prefetch to a patch appset:.....	138
Modifying a Patched Appset	138
Extracting an Application for Patching.....	139
To extract an application from an appset:.....	139
Chapter 10: Troubleshooting	141
Identifying Common Problems.....	142
Solutions to Common Problems	142
Missing Shortcuts.....	142
Blank Shortcuts	143
File Associations Do Not Work.....	143
Application Fails to Launch	143
Printing Problems	143
Fonts Do Not Appear	143
Folder or Files Disappear from the Virtualized Application	144
Hide "Add/Remove Programs" Entry	144
Roaming and Folder Redirection Issues.....	144
Prerequisites.....	144
Unique User Accounts	145
Service Show as Running and Cannot Be Stopped	145
Windows Services or Drivers Do Not Start.....	145
DCOM and COM+ Services Not Remotely Accessible	146
Plug and Play Device Drivers	147
Alternative Data Streams	148
Anti-virus and Firewall Applications.....	148
Application Copy Protection	148
Approach to Resolving Other Problems.....	148

Appendix A: Folder Template Mapping	149
Folder Mappings.....	149
Appendix B: Editing Configuration Files	153
Conventions.....	153
preloadrehives.dat.....	153
Example.....	153
procexcluded.dat.....	153
filefilt.dat	153
Example:.....	153
regfilt.dat.....	154
procfilt.dat.....	154
defprotsel.dat.....	154
Example:.....	154
fileexcluded.dat	155
Examples:.....	155
regexcluded.dat.....	155
regdifferences.dat.....	156
rlfnfilt.dat.....	156
appratng.dat	156
Appendix C: Command Line Usage	157
Syntax	157
Commands.....	157
Glossary.....	159

Introduction

Cloudpaging Studio converts Windows applications into a Cloudpaging container without requiring any source code. Cloudpaging Studio gives you:

- Flexibility to control how the application is integrated with the Windows desktop, ensuring a better end user experience.
- Full control over files, registry settings, and how additional requirements, such as services, are to be handled.
- Drivers, libraries and additional dependencies can be included in the Cloudpaging container.

All that is required to package Windows application is a desktop running Windows, preferably a virtual machine, and Cloudpaging Studio to encode the application into a container.

This area includes the following:

- Features of Cloudpaging Studio
- Guide Conventions
- Getting Support

Features of Cloudpaging Studio

- Simple to use application installation monitoring tool, enabling Microsoft Windows installers to produce an equivalent container.
- Advanced virtualized application delivery solution supporting integrated, isolated application virtualization.
- Custom application event handles, allowing scripts to be attached to key application triggers, such as activation, start, close, deactivation, etc.
- Local execution mode, allowing a packaged application to be tested without the need to install on the server.

Guide Conventions

The following typefaces are used throughout this content:

Bold text	Names of windows, panes (portions of a window), dialog boxes, menus, commands, buttons, predefined folders, and keyboard keys. Examples: Click OK . Press Enter . Navigate to the My Documents folder.
<i>Italic text</i>	Indicates references to a section within this document or other documentation. Also used to emphasize new terms. Example: Refer to the <i>Cloudpaging Web Portal Integration Supplemental Guide</i> for further information.
> symbol	Sequence in which you should select a menu option. Example: "File > New > Document" means "click the File menu, click New, and then click Document."
1, 2, 3	Numbered items indicate sequential steps in a procedure. Sub steps are indicated with indented bullets. Relevant graphics or screen captures may follow steps.
Screen captures	The screen captures in this document are examples only. They may not exactly match the user interface on your system.

The following formats are used throughout this document:

NOTE

This block style identifies additional information about the preceding text, or the text immediately to follow.

Important

This block style identifies information about actions that might cause problems with an application, your local PC system, or your data, including data loss.

Getting Support

Website

Support information is available online on at: <http://support.numecent.com/>

Email

You can request technical support by sending an email to support@numecent.com. Our support staff will address your email questions in a timely manner.

Chapter 1: Cloudifying Overview

Cloudifying applications is the process of converting a software application to a format that can be paged to **Cloudpaging Player** on a Windows desktop. To package, or cloudify, an application, the target application is first installed on a Windows desktop that is dedicated to cloudifying and has **Cloudpaging Studio** installed.

Cloudpaging Studio determines how the installed application works in the Windows environment and packages the required components and configuration information into a compressed, encrypted file called an appset. The appset contains all the files, folders, and registry settings needed to reproduce the application on the user's PC. The appset is loaded on a **Cloudpaging Server** at which point the target application becomes available for users.

Cloudpaging Studio controls what minimum instructions in the application should be available the first time the application is deployed to the user. These minimum instructions will be demand-paged first, then the application will run. This saves time and lowers bandwidth required for users to ensure even large applications can run right away. Additional application features are seamlessly requested from the **Cloudpaging Server** automatically.

The following are explained in this chapter:

- Cloudifying Environment
- Target Audience
- Recommended Skills and Knowledge
- Data Needed to Cloudify an Application
- Names for a Cloudified Application
- Core Concepts
- The Cloudpaging Studio UI
- Cloudifying Phases
- Repackaging, Modifying, or Patching Appsets
- Reusing Project Files
- Appset File Structure

Cloudifying Environment

To transform an application into a cloudified application, at least two PCs are required: a Cloudifying PC and a Target PC. Depending on the operating system requirements of the intended users, the Cloudifying and Target PCs need to have Windows 10 or later operating system installed.

A “clean” PC is needed each time an application is cloudified, having nothing but the following installed: the OS, **Cloudpaging Studio**, all the most recent drivers, any additional programs that the target application needs in order to function properly. A virtual machine (VM) is recommended for efficiency and a copy of the “clean” VM can be saved and reused for each cloudifying operation.

NOTE

The Cloudifying PC should also have a minimum screen resolution of 1024 x 768 to accommodate the Cloudpaging Studio interface.

Target Audience

This content is written for a technician or engineer who will have the role of *Appset Designer*, the person responsible for using **Cloudpaging Studio** to change Windows-based applications into a cloudified format.

Recommended Skills and Knowledge

The following list is meant as a guideline for recommended skills and knowledge in Information Technology (IT), PC platforms and Operating Systems (OS) technologies for an *Appset Designer*, the person who uses **Cloudpaging Studio** to cloudify applications.

- Working knowledge of Windows platforms 7 and later, the differences between the platforms, and the installation of service packs.
- Working knowledge of hard disk partitions and renaming.
- Working knowledge of basic, distributed and networking file system aspects of the Windows OS, including the organization of Program Files, Roaming user profiles, User start menus, and special folders.
- An understanding of DLLs, program executables, shared libraries, and command-line parameters.
- Working knowledge and understanding of the Windows registry, including machine profiles, user profiles, and software registry entries.
- Working knowledge of setting up and configuring Windows applications, such as Microsoft Office.
- Understanding of application installers and their packaging (e.g., InstallShield AdminStudio) will be helpful.

NOTE

To optimize the performance of each cloudified application, it is recommended that an experienced user of the target application participate in the cloudifying process. Those who have packaged applications with other products like Microsoft App-V will be able to accelerate quickly to proficiency with Cloudpaging Studio.

Data Needed to Cloudify an Application

An application is made up of various components such as files and registry settings called *assets*. To cloudify an application, it is necessary to collect and store all assets needed to run the application, along with configuration

information that describes how those assets will be processed and managed when paged to **Cloudpaging Player** on users' PCs. **Cloudpaging Studio** stores the assets and configuration information in a *cloudifying project* (or simply *project*).

The following sections discuss assets, configuration information, and projects in greater detail.

Application Assets

Assets are items needed to cloudify an application. An application may have various types of *assets*. The most common are files, registry entries, and environment variables. There are also more specialized types of assets such as fonts, startup items, services, and drivers that are themselves made up of one or more file and registry entries.

The assets of an application are collected by **Cloudpaging Studio** into a project during application installation. The appset designer can later add more assets to the project or remove some, as needed.

Configuration Information

In addition to storing an application's assets, it is also necessary to keep configuration information that describes the application and determines how its assets will be processed and managed when paged to **Cloudpaging Player** on the end-user device.

The appset designer makes configuration choices based on the anticipated way that an application will be used or based on special application needs.

The Cloudifying Project

To cloudify an application, the appset designer must first create a *cloudifying project* (or simply *project*) that will contain all assets and configuration information needed to produce the *cloudified application*, or *appset*. The project contents are saved to a *project file* (**.stw**), also called an *STW file*, for later retrieval. Assets are added to the project through the application *capture* process. During the capture process, the application installer is run and **Cloudpaging Studio** monitors changes made to the system. Specifically, it keeps track of files and registry values that are added to the system or modified, along with their folder and registry key locations. At the end of the process, **Cloudpaging Studio** adds to the project the tracked assets that can then be fine-tuned by the appset designer, along with the configuration information, if needed.

Finally, the project is used to produce an appset through the *cloudifying* process.

Names for a Cloudified Application

There are three different names for a "cloudified application" depending on the point of view (POV) of the person using it:

- Appset Designer's POV - "appset"
- Cloudpaging Server Admin's POV - "application"
- User's POV - "cloudified application"

After a cloudified application has been paged, it is referred to as a "**paged application.**"

Core Concepts

Virtualization

Application virtualization allows for the application to be paged to a user's PC while protecting the operating system from resource conflicts and environment corruption that can occur with traditional installation methods. Virtualization creates a boundary between the application resources and the local system resources.

Cloudpaging Studio takes virtualization to a new level by making virtualization configurable.

Virtualization is a **Cloudpaging Studio** feature that makes it possible to assign a specific disposition layer to individual application *assets*, such as files, folders, registry keys, and registry values. The layer determines whether the system resource can be seen by the local system, and whether it is permanent or can be removed.

For certain resources, such as fonts and environment variables that are typically shared by the entire system, a special method called *injection* is used to maintain the isolation layer. There are four *Disposition Layers*:

- **Installed-permanent (Layer 1)** - Copies assets (files, folders, registry keys, and registry values) permanently onto the local system, and can be seen by the entire local system.
- **Installed-temporary (Layer 2)** - Installs assets during the activation process, and uninstalls assets during the deactivation process. If the asset already exists on the local system, the original asset is backed up before the new asset is installed, and when the new asset is uninstalled the original asset is restored.
- **Virtual-integrated (Layer 3) - (default setting)** Assets that can be seen both by the paged application and the local system, but are not physically installed on the local system.
- **Virtual-isolated (Layer 4)** - Assets that can only be seen by the paged application, and are not physically installed on the local system.

Prefetching

Prefetching is the process of fetching application pages prior to their being explicitly requested by **Cloudpaging Player**, in anticipation of being needed by the application. Unlike conventionally installed applications, application paging does not require all application files to be always available. Instead, only some key pages from the ASP's server are necessary for the application to launch and perform basic functions.

The prefetch file is created by logging pages that are being fetched when the application is executed. This includes all the functions required by the application when the user opens it for the first time. During the optimization phase of the cloudifying process, you can capture three types of Prefetch files:

- **Stage-1 Prefetch** - This group of application pages are those that are needed to start the application, as well as to perform typical actions for the first few minutes of using the program, such as opening files, saving files, etc. The process of fetching these pages from the server to **Cloudpaging Player** always happens before an application launches. **This stage is always recommended for best application performance.**
- **Stage-2 Prefetch** - This group of data consists of application pages that are needed to perform other important tasks the user is likely to perform. The amount of captured Stage-2 Prefetch pages depends on the trade-off among several factors such as the time it will take to fetch these pages, the cache size required to store them, etc. Examples of Stage-2 Prefetch are using various functions of a business application and loading the second level of a game. **Cloudpaging Player** will fetch these pages slowly in the background after the application launches. This stage is optional.

Compression

Compression of an appset decreases the file size, and the smaller it is, the faster the appset pages. Application paging operates on file pages as units of file contents. Each application file to be cloudified is divided into a series of pages stored in a content (STC) file by **Cloudpaging Studio**, published to **Cloudpaging Server**, transmitted by **Cloudpaging Server**, and finally received and virtualized on the local PC by **Cloudpaging Player**. Since file pages are passed through many components normally residing on different machines, the pages of data should be compressed to make data transmission and storage efficient.

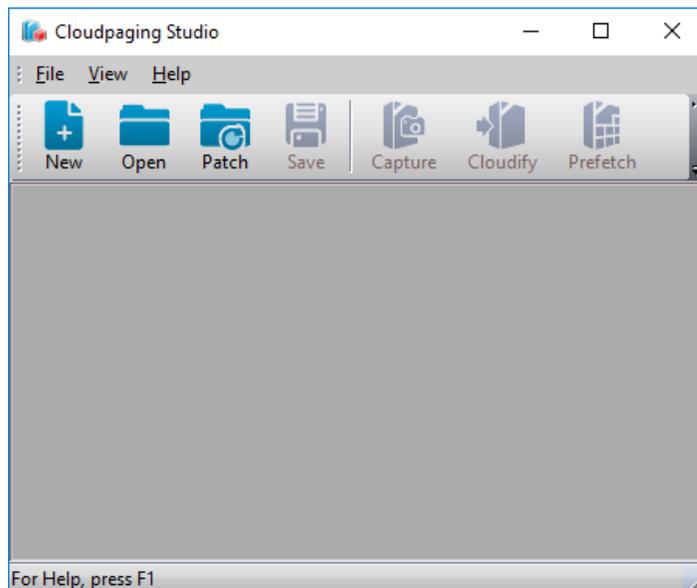
Encryption

Encryption of appsets provides a high degree of security, ensuring that the contents of the appset remains safe and private when paging over the Internet. Each page in the STC file is individually encrypted, which protects the content during transport and when cached. As with compression, since the pages of data can reside on various machines, you should always encrypt the content.

The Cloudpaging Studio UI

The Main Window

The **Cloudpaging Studio** UI consists of a *main window* containing a *menu bar*, a *toolbar*, and a *status bar*, each of which can be individually hidden. The rest of the window, between the toolbar and status bar, is called the *common area*; it is initially blank when **Cloudpaging Studio** starts up.



The toolbar contains general commands that reflect the typical cloudifying workflow.

The status bar displays a brief description of a command when the mouse cursor hovers over a menu item or toolbar command. It also shows the state of the **Caps Lock**, **Num Lock**, and **Scroll Lock** keys. It may show other context-specific information as well.

The common area typically hosts one or more workspaces in a tabbed fashion once a workspace is loaded or created (see [Workspaces and Views](#)). Each workspace, in turn, hosts one or more views in a tabbed fashion as well. Additionally, the common area may host other special windows such as the **Find Results** window (see [Other Windows](#)).

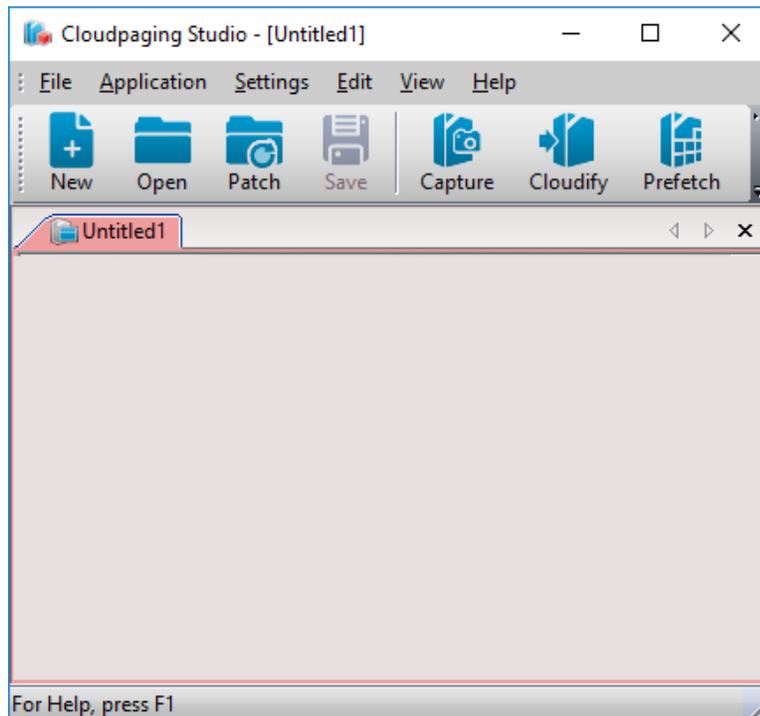
Workspaces and Views

When the appset designer creates a new project or opens an existing project file, a *workspace* is created for the project. The workspace is the area of the main window where the appset designer can configure the project's settings, manage its assets, and cloudify an application. By default, it occupies the entire common area of the main window.

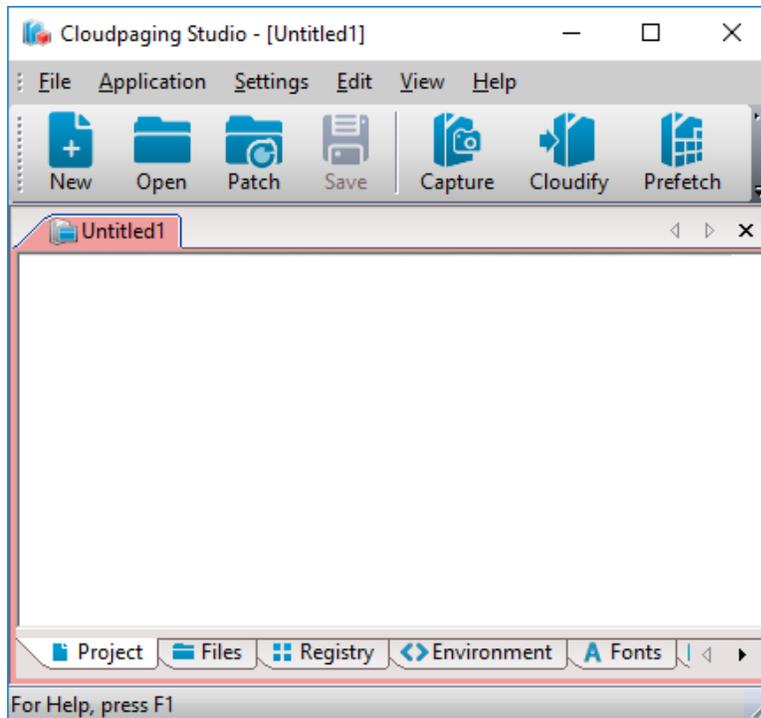
NOTE

There are other types of workspaces that are associated with the different types of files that Cloudpaging Studio can open. In addition to project files, Cloudpaging Studio allows opening AIB files and token files, which are appset-related files discussed in [Appset File Structure](#). These files get opened in an AIB workspace or token workspace, respectively (see [The AIB Workspace](#) and [The Token Workspace](#)).

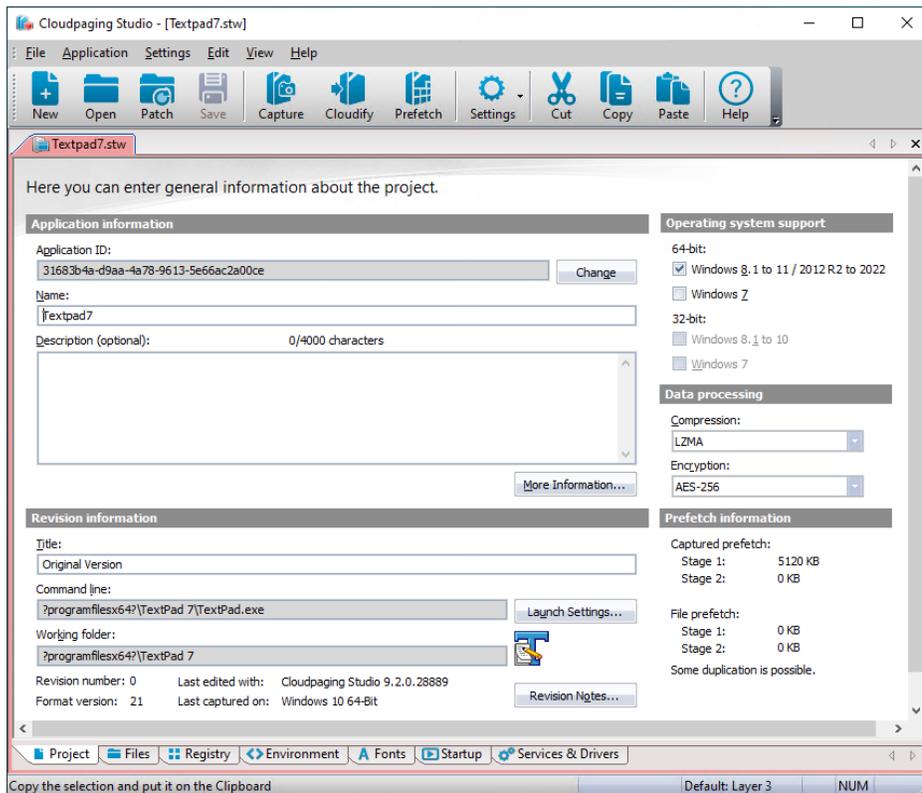
Multiple workspaces (possibly of different types) can be open simultaneously. Workspaces are tabbed at the top to make it easy to switch between them by clicking their corresponding tabs. The workspace where the appset designer is currently working is the *active workspace*.



A workspace is made up of one or more *views* depending on its type. Views are used to organize a workspace to make it easier for an appset designer to manage its contents. Each view is used to manage a different type of content or, sometimes, to present the same content in different ways. Views are tabbed at the bottom within the workspace area so it is easy to switch between them by clicking their corresponding tabs. The view that is currently selected is the *active view* and it occupies all the available workspace area.



The figure below shows the main window with an open project:



The following sections discuss each type of workspace and the views they contain.

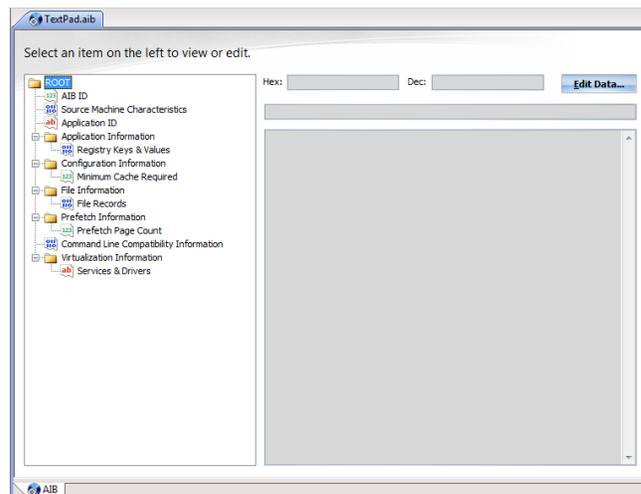
The Project Workspace

Cloudpaging Studio creates a project workspace when a new cloudifying project is created or an existing project file is opened. It contains the following views:

- *Project view* (see [Enter Basic Application Information](#))
- *Files view* (see [Modifying Files and Folders](#))
- *Registry view* (see [Modifying Registry Entries](#))
- *Environment view* (see [Modifying Environment Variables](#))
- *Fonts view* (see [Modifying Fonts](#))
- *Startup view* (see [Modifying Startup Items](#))
- *Services & Drivers view* (see [Modifying Services and Drivers](#))
- *Short view* (see [Modifying Shortcuts](#))

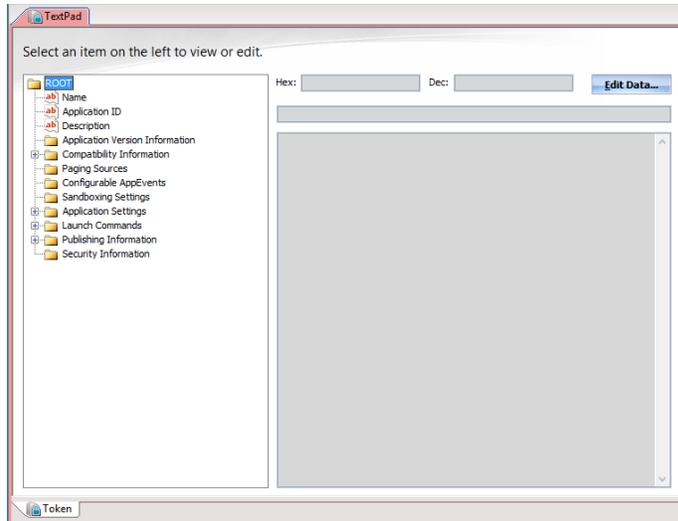
The AIB Workspace

An AIB workspace is created when an existing AIB file is opened for editing (see [Editing an AIB file](#)). It contains a single *AIB view* as shown in the figure below.



The Token Workspace

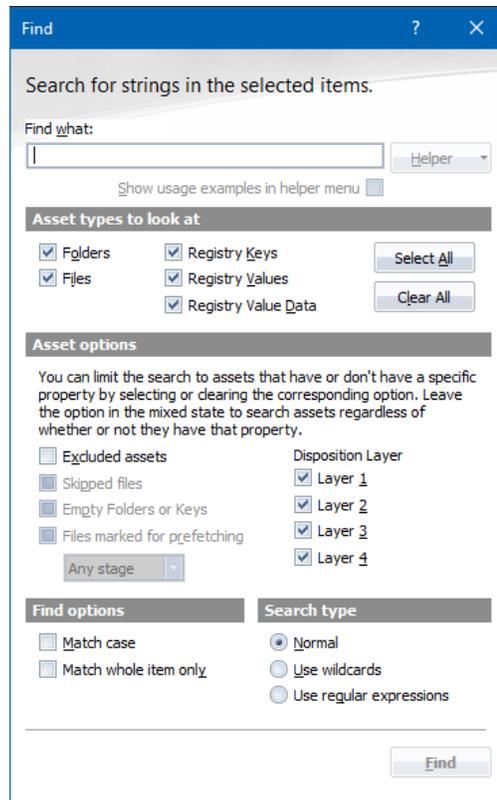
A token workspace is created when an existing token file is opened for editing (see [Editing Token files](#)). It contains a single *Token view* as shown in the figure below.



Other Windows

A workspace may contain additional special-purpose windows that appear alongside the views in the workspace area, such as the **Find Results** window (project workspace only). However contrary to views, which are always visible, these special windows may be shown or hidden as desired.

Other windows called *dialogs* appear on top of the main window when certain commands are invoked to obtain user input. Most **Cloudpaging Studio** dialogs are *modal* and must be dismissed before it is possible to interact with the main window again. Others (such as the **Find** dialog) are *modeless* and allow switching between them and the main window.



Cloudifying Phases

As illustrated below, there are **four phases** that are required to cloudify an application using **Cloudpaging Studio**. Each cloudified application can later be added to a **Cloudpaging Server** and paged to **Cloudpaging Player** installed on users' device or VDI VM.

This section presents an overview of the four cloudifying phases. Each phase is covered in detail in a subsequent chapter.

- Preparing the Environment
- Capturing the Application Installation
- Cloudifying the Application
- Testing and Optimizing the Appset

Phase 1: Prepare the Environment

See [Chapter 2, Preparing the Environment](#).

Cloudifying applications with **Cloudpaging Studio** requires at least two (physical or virtual) PCs: A *Cloudifying PC* with **Cloudpaging Studio** installed, and one or more *Target PCs* with **Cloudpaging Player** installed, for testing. The desired operating system, all necessary drivers, and any prerequisite software (such as the most recent OS service packs or .NET) should be installed on all PCs.

If the appset is intended to run on multiple operating systems, then you should always cloudify on the earliest OS version. For example, if the application works on both Windows 7 and Windows 10, then it is best to cloudify the application on Windows 7. If the application works on Windows 7 32-bit and 64-bit, then it is best to cloudify on Windows 7 32-bit. Cloudifying on the earliest OS typically has the best chance of working because that OS will have less dependencies already installed as opposed to later OS versions. This same general rule applies to OS service packs.

Important

It is not always possible to have a single appset work on all target operating systems. In this case multiple appsets may be required for the different operating systems.

If physical PCs are used for the cloudifying, an imaging utility such as *Ghost* from Symantec should also be installed so all PCs can be easily restored to their pre-cloudifying state. For maximum cloudifying efficiency, however, Virtual Machines (VMs) are recommended.

To optimize the performance of each cloudified application, it is recommended that an experienced user of the target application participate in the cloudifying process. Otherwise, the appset will still function but maybe not as optimally as would be possible.

Phase 2: Capture the Application Installation

During the capture process, the application installer is run and **Cloudpaging Studio** monitors system changes (files and registry values that are added to the system or modified). Each time you capture the installation process of an application, it is necessary to start with a Cloudifying PC and Target PCs that are in their pre-cloudifying state, as described in Phase 1. If the capture process is interrupted for any reason (such as a system reboot after the installation is complete) the project file allows the capture process to resume.

See [Chapter 3, Capturing the Application Installation](#).

Phase 3: Cloudify the Application

Configure project settings – From the Project view and Settings dialog of the **Cloudpaging Studio** UI, the following project information include:

- Application Information
- Supported Operating Systems
- Data Compression Method
- Data Encryption Method
- Configurable AppEvents Settings
- Virtualization Settings
- Sandboxing Settings
- Security Settings

See [Chapter 6, Cloudpaging Settings](#).

Modify File, Registry, and Environment Content and Settings – As required by the particular application, you can modify Files, Registry, and Environment views within the Cloudpaging Studio UI.

See [Chapter 7, Modifying Project Assets](#).

Cloudify the Application – After the cloudifying process is complete, the result is an appset. The *cloudified application* is called an appset from the point of view (POV) of an Appset Designer, the person who uses Cloudpaging Studio to change Windows-based applications into a cloudified format. (A *cloudified* application is called a cloudified application from the users POV, and an application from the Cloudpaging Server admin' POV). The appset (.stp file) is the outer layer that encapsulates the other three application components: the STC file (.stc – the code and other configuration information that comprises most of the application), the AIB file (.aib – metadata that describes the content of the STC file), and the Token file (comprised of security data). Later, after the final appset is created, it is then published to the Cloudpaging Server and becomes available for Cloudpaging to users.

Phase 4: Test and Optimize the Appset

Test the Appset. Testing the appset includes loading it onto a Target PC with **Cloudpaging Player** installed and running the application. Local paging, the process of paging the appset to **Cloudpaging Player** on a single Target PC, allows you to test several appset iterations quickly, without publishing the appset to **Cloudpaging Server**.

NOTE

Local paging an appset can only be used for 30-days after the application is packaged. Please use Cloudpaging Server to test after 30-days. Identify and Create Prefetch Files. After installing Cloudpaging Player on the Target PC, you will need to enable the prefetch Capture button on the Cloudpaging Player toolbar, which includes Stage-1 and Stage-2 options. With Stage-1 selected, you will then click Capture to start capturing the Stage-1 prefetch, perform the desired actions on the target application, and click Capture to end the capture process. Then, if you choose to add a Stage-2 prefetch, you will select Stage-2 and repeat the same process.

Set Compression. During the testing phase, compression is usually set at “None” in order to speed up the cloudifying process. After testing is complete, you can change compression to “LZMA” for the optimal delivery speed.

Set Encryption. Again, during the testing phase, encryption is usually set at “None”, in order to speed up the cloudifying process. After testing is complete, you can change encryption to “AES-256 Enhanced” for the optimal security.

Save the Project file. This will be required to patch the appset in the future.

Publish the appset to Cloudpaging Server. Once optimized, the appset is ready to be published from **Cloudpaging Studio** to the **Cloudpaging Server**, which is done by copying the final STC file to the **Cloudpaging Server**. After the **Cloudpaging Server** admin completes necessary server configurations, the application is ready to be paged to the **Cloudpaging Player** running on end-user devices.

See [Chapter 5, Testing and Optimizing the Appset](#).

Repackaging, Modifying, or Patching Appsets

Cloudpaging allows for a number of different methods to deploy required changes to applications. There can be many reasons why changes to the application are needed, such as to fix a deployment issue, to include additional restrictions, or for new version updates or patches. Cloudpaging has controls to allow for the following type of deployments:

Re-packaging an Appset

Many issues found with the application can simply be fixed by re-packaging. This can include making modifications such as adding, deleting, or changing files or registry keys. The changes can be easily done using the various view tabs in **Cloudpaging Studio**. The appset can then be re-packaged and re-tested prior to deployment. While re-packaging can take time to complete, it is useful for changes that involve the file system structure.

See [Chapter 7, Modifying Project Assets](#) for more information on modifying the project assets prior to repackaging.

Modifying an Appset

The TOK and AIB inside an appsets can be modified when small changes are required, without needing to modify the application file structure. The purpose of modifying an appset is to change security settings, disposition, prefetch settings, registry settings, configurable appEvents, drivers and services, file attributes, as well as file names and locations. Modifying an appset avoid the potential time overhead of re-packaging.

NOTE

Modification are not stored in the project file and will be lost if the appset is ever re-packaged or patched.

See [Chapter 8, Modifying an Appset](#) for more information on modifying the appset without repackaging.

Patching an Appset

If an application has a new version available, then patching the appset is the best option. Patching is best used when there are minor changes to be made to an appset and it is desired to maintain the previous version. For example, a patch or hotfix for an application can be added to an existing appset as a patch (e.g., Photoshop 5.5.1 updates Photoshop 5.5.0). If the application update contains major of a large number of changes, then a new appset is recommended.

See [Chapter 9, Patching an Appset](#) for more information.

Packaging a New Appset

With a major new version of the application, it is better to generate a new appset due to the potentially large number of differences between two versions (e.g., Photoshop 6.0 supersedes Photoshop 5.5). An appset can

contain multiple patch versions allowing for easy deployment or rollbacks for each individual patch number. When patching, changes to the appset configuration (e.g., CAE's, Virtualization settings, Sandboxing settings, etc.) are not allowed. To modify these configurations the appset must be modified or re-packaged.

How To Deploy Updated Appsets

Cloudpaging fully supports both provisioning of new appsets, patched appsets, and modified or repackaged appsets. The **Cloudpaging Server** allows for applications to be upgraded using a major or minor deployment method.

- **Minor Upgrade** - A minor upgrade is used to deploy patched versions of an application. Minor version will retain the application settings.
- **Major Upgrade** - A major upgrade is used to deploy a new appset, modified appset, or repackaged appset.

See the Cloudpaging Server Administration Guide for more information.

Reusing Project Files

Beginning with version 9.0 of **Cloudpaging Studio**, you can now reuse project files (*.stw) on a computer which was not the original computer used to package the appset. This will enable other users to leverage the work done to cloudify a specific application without having to repeat the same steps.

You will also be able to share and download project files posted online.

Requirements

In order to use a project file created on a different computer:

- Both computers must have the same version of Windows installed. In other words, Windows 10 users should only use project files created on other Windows 10 machines.
- Windows can be installed to a different drive letter and you can redirect default folders (Documents, Pictures, etc.) to different locations.
- The project file must be for the same version of the application. This means that the installers used must have MD5 checksum on both computers.
- You must have the required files for any Configurable AppEvents (CAEs) defined in the project file.
- Your computer must meet the same requirements for dependencies (.NET version, Runtimes, etc.) as the source computer.
- If the application depends on specific hardware, you must have that hardware installed or connected to your computer.
- The application being cloudified cannot hard-code any file paths for its configuration files.

Appset File Structure

An appset is made up of multiple files. The primary files described below are always present whereas the additional files may or may not be in the appset depending on the configuration settings of the project used to create the appset. All file types are shown with their corresponding file extensions in parentheses.

Primary files

STP or appset file (.stp)

Compressed file that contains all of the other files below. It can be uncompressed by changing the extension to .zip and unzipping the file.

STC (.stc)

Contains all the data pages of the application being cloudified. This file is the largest portion of the STP file.

AIB (.aib)

Contains all the metadata for application files and registry keys.

TOK or token file (.tok)

Contains the security data and general information about the application.

Additional files**ICO or icon file (.ico)**

Contains the graphic file for the application icon provided by the software vendor (example, Word icon for each Word file).

CAE or Configurable AppEvent file (.cae)

There is one .cae file per AppEvent trigger and each can initiate an action such as running a VB script that installs other needed applications.

HTML (.html)

End User License Agreement (EULA) for the application.

META-INF (file folder)

The META-INF file contains one or more subfolders, each of which contains a copy of the STW project file for each patch version. The original project file is contained in the Patch-0 folder. The file for each subsequent patch is located in Patch-1, Patch-2, etc.

XML (.xml)

Readable information about the packaged application, such as the name and App GUID. These values are read-only to provide easy, quick access without requiring Cloudpacing Studio to view the package.

Chapter 2: Preparing the Environment

As described in *Cloudifying Overview* chapter, application cloudifying with **Cloudpaging Studio** has four phases.

Phase 1: Preparing the Environment includes procedures for preparing a clean cloudifying PC, preparing the testing PC(s), and the procedure for installing **Cloudpaging Studio**.

The following are explained in this chapter:

- Cloudifying PC Technical Requirements
- Target PC Technical Requirements
- Preparing a Clean Cloudifying PC
- Installing Cloudpaging Studio

Cloudifying PC Technical Requirements

Two or more PCs -You will need one Windows-based PC dedicated to cloudifying applications (*Cloudifying PC*) and one or more Windows-based PCs to test the cloudified applications (*Target PCs*).

The Cloudifying PC must meet or exceed the system requirements of the application being cloudified.

- **Operating System** - One of the following:
 - Window XP (32-bit)
 - Windows 7 (32-bit or 64-bit)
 - Windows 10 (64-bit)
 - Windows 11 (64-bit)
 - Windows Server 2003 (32-bit)
 - Windows Server 2008 R2 (64-bit)
 - Windows Server 2016 (64-bit)
 - Windows Server 2019 (64-bit)
 - Windows Server 2022 (64-bit)
- **Hardware** - The following requirements have been established based on acceptable compression and encryption performance:
 - **CPU:** 3.0 GHz or faster
 - **RAM:** 4GB or greater
 - **Disk Space:** 60 GB or greater (estimated), depending on the installation size of the application
 - to be processed.
- **Cloudifying Software - Cloudpaging Studio.** An “administrator” privilege level is required to install and create appsets, although not to modify existing appsets.

Recommended Software - Virtual Machine software (such as *Virtual Box®* from Oracle) or, for physical machines, Disk Imaging software (such as *Ghost®* from Symantec®)

Target PC Technical Requirements

- **Test Application Software - Cloudpaging Player** (Please refer to the *Cloudpaging Player User Guide* for additional requirements).
- **Support Software** - As required by the paged application. Depending on the application, specific video card drivers (e.g., DirectX) may be required. See the application manufacturer’s website for current updates.
- **Additional Software** - Any other software as required to best reflect the typical system for targeted users. For example, Microsoft Office for users in an office environment.

About Cloudifying Performance

The cloudifying process is both I/O and CPU intensive. Performance measurements have shown that the following criteria have the greatest effect on cloudifying speed, in increasing order:

- Physical hardware (>30% speed improvement over a virtual machine)*
- SCSI instead of IDE hard drives
- Faster front side bus
- Faster processor
- Dual/Quad processor instead of single processor (>10% speed improvement)*

* These measurements are provided only as a reference. Actual results may vary based on your hardware and software system configuration.

NOTE

PCs can be physical computers or virtual machines (VMs). While physical machines are faster, the overall cloudifying process is quicker using VMs.

Preparing a Clean Cloudifying PC

A “clean” PC is needed to cloudify an application (*Target Application*). As listed in *Technical Requirements* above, a clean PC consists of the OS, **Cloudpaging Studio**, and for physical PCs, an imaging program like Symantec’s Ghost™.

Uninstalling extraneous applications is typically not sufficient, since unwanted *artifacts* (DLLs, etc.) are often left behind. Not removing other applications and artifacts could cause conflicts during the cloudifying process, cause extra data to be included, and cause other conditions that prevent the process from functioning correctly.

To prepare a clean Cloudifying PC, do the following:

- 1) Format the disk drive and install a supported operating system. See [“Cloudifying PC Technical Requirements”](#) **Make sure that you have the manufacturer’s Recovery Disk or Original Driver Disk when doing this step.**
- 2) Install any needed drivers and the latest OS service packs. Driver discs for various system components such as the network card, video drivers, etc. should come with the PC. However, you may have to download the drivers from the Internet or your company server.
- 3) Verify that no firewall, anti-virus, or anti-malware software is installed on the Cloudifying computer. **Cloudpaging Studio** will capture active processes during installation. This will include real-time scanning anti-virus and anti-malware software. Microsoft anti-malware software should be disabled if it can’t be uninstalled. These include:
 - Microsoft Anti-Malware
 - Microsoft Security Essentials
 - Windows Defender
 - Microsoft Customer Experience Improvement Program
 - Windows Firewall
- 4) Install **Cloudpaging Studio** on the cloudifying PC. See [Installing Cloudpaging Studio](#) for complete instructions.
- 5) Create a duplicate of the clean PC.
 - **For a Virtual Machine:** Take a snapshot of the VM, which can later be reverted or cloned.
 - **For a physical PC:** Save an image of the Cloudifying PC configuration using a disk imaging utility such as Symantec’s Ghost™. This process consists of copying the entire PC configuration to a separate or disk partition for later reconfiguration of the same or another PC.

NOTE

Virtual PCs (VMs) are recommended for both the Cloudifying PC and Target PCs over physical PCs, because a clean VM can be copied and therefore can provide the fastest recovery of a clean OS. While a physical PC can be faster for the last cloudifying phase, virtual PCs are faster for the initial phases, and are faster for the cloudifying process as a whole.

Installing Cloudpaging Studio

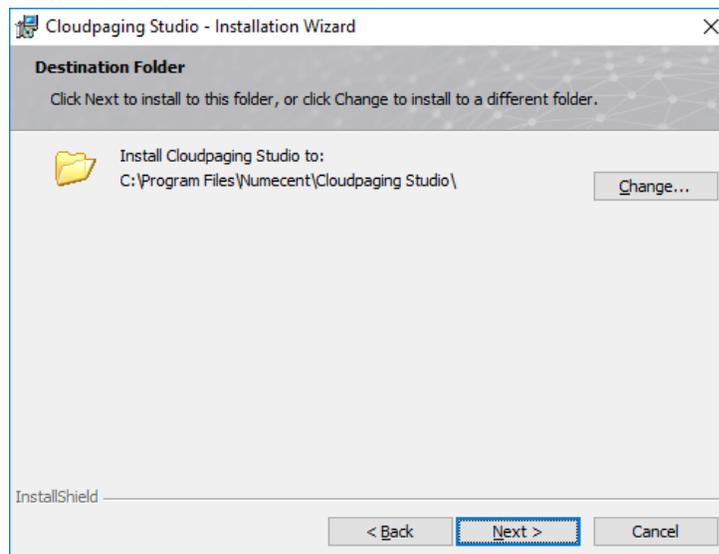
After you have completed the environment preparation, follow these steps to install **Cloudpaging Studio**:

To install Cloudpaging Studio:

- 1) Run the Cloudpaging Studio installer: Cloudpaging Studio.msi. The Cloudpaging Studio - InstallShield Wizard dialog appears.



- 2) Click Next. The License Agreement dialog appears.
- 3) Read the license agreement, select **I accept the terms in the license agreement**, and click **Next**. Otherwise, click **Cancel** to cancel the installation.



The **Destination Folder** dialog appears.

- 4) Specify the directory into which you want to install **Cloudpaging Studio** and click **Next**, or click **Back** if you want to change previous settings. The **Ready to Install the Program** dialog appears.
- 5) Click **Install**, and then click **Next** on the subsequent dialogs. The **Installing Cloudpaging Studio** dialog appears.
- 6) Click **Finish**. The installation has completed. The **Installation Wizard Completed** dialog appears.

Chapter 3: Capturing the Application Installation

As described in *Cloudifying Overview* chapter, application cloudifying with **Cloudpaging Studio** has four phases. **Phase 2: Capturing Installation** includes all the steps pertaining to capturing the installation of the target application and installing any associated applications that are needed for the target application to function properly.

NOTE

Regarding the capture phase of the cloudifying process, even though the program may require a specific version of DirectX, Acrobat, or QuickTime; these programs/plugin-ins are not needed for the processing system. They will need to be installed onto the testing system.

The following are explained in this chapter:

- Procedure for Capturing the Application Installation
 - Create a new Project
 - Enter Basic Application Information
 - Start the Application Installation Capture Process
 - Install the Application
 - Run the application (optional)
 - Capturing System Reboots
 - Stop the Installation Capture
 - Enter Remaining Application Information
 - Save the Project
- About OS Compatibility Settings

Procedure for Capturing the Application Installation

Step 1: Create a new Project

Click **New** on the **Cloudpaging Studio** toolbar. A new project is created that can be saved to a project file (**.stw**). The project file is where all cloudifying configuration settings and assets are stored.



NOTE

You should clear the Log file for each new project since the log file is appended to each time. This can be accomplished by clicking the Application menu and then select Clear Log.

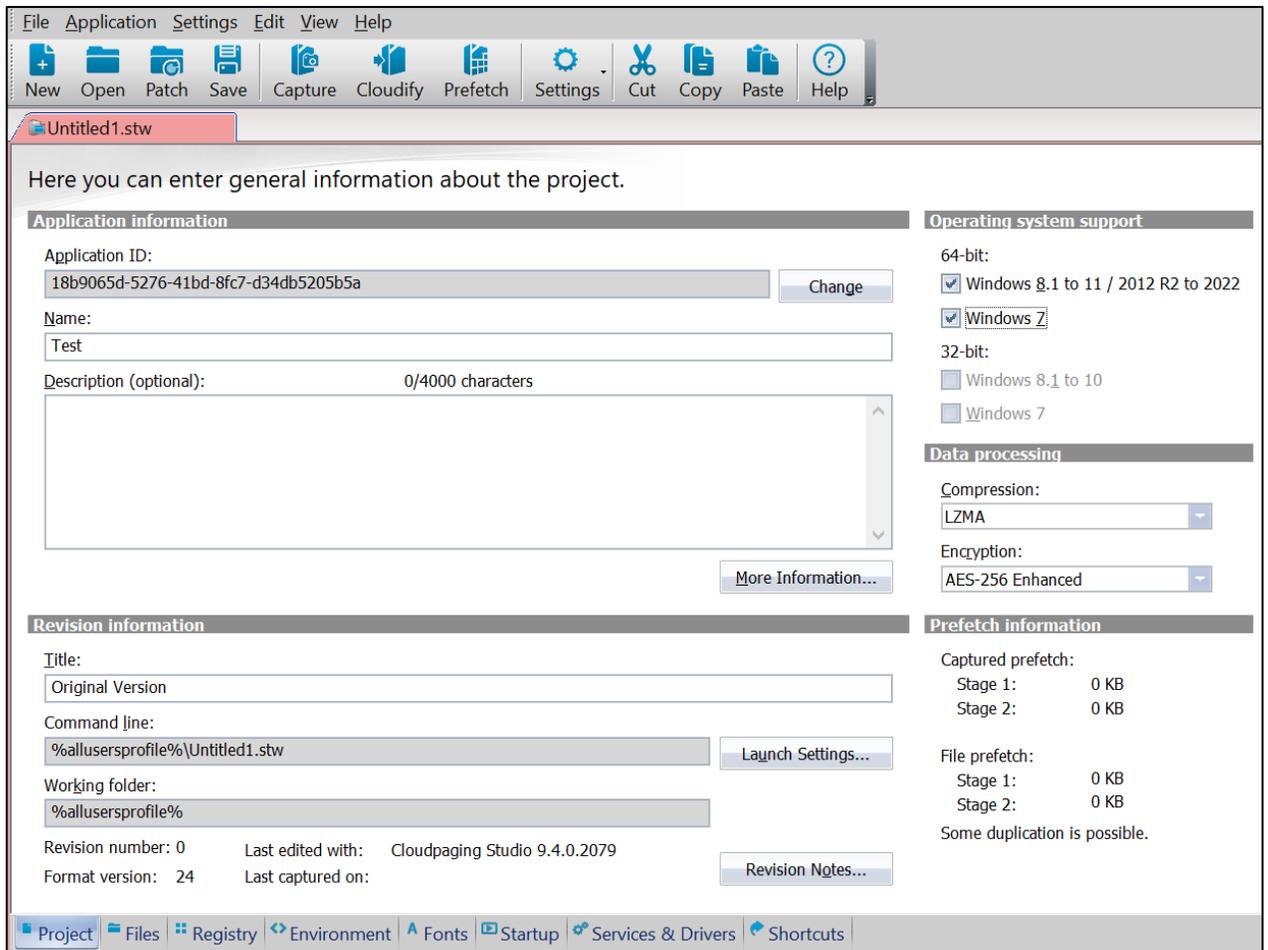
Step 2: Enter Basic Application Information

On the **Project** view of the **Cloudpaging Studio** UI, enter the following information, which is typically printed on the software cloudifying or in other materials provided by the publisher:

- Application Information
- Application Description
- System Requirements
- Operating System Support

Application Information

- **Application ID** – This is a unique identifier that is newly generated each time you click **Change**.
- **Name** – This name is displayed in the **Cloudpaging Player** Application pane.
- **Application Description** – This field describes the capabilities and uses for the software program. This description is used primarily for informational purposes and can be shown on the **Cloudpaging Server**.



Operating System Support

The Operating System Support dialog specifies the Operating Systems for which the application is compatible. **Cloudpacing Player** will use this information to determine if the appset is compatible with user's Operating System. The Operating System compatibility information is usually included with the product information printed on the application box or on the publisher's web site, or is determined through testing.

NOTE

If multiple OS version are desired as the Target OS, then you should always cloudify on the earliest version. For example, if an application supports Windows 7 or later, then it is best to cloudify on Windows 7. The same is recommended for service packs.

If the processed application is run on a machine for an unsupported Operating System, **Cloudpacing Player** will display a dialog indicating that the Operating System is not supported and will not start the application. Sometimes testing will indicate that an application is compatible with an Operating System not listed in the application's system requirements.

Selecting the OS options sets the default operating system(s) for which the target application is compatible. The Admin service has the capability to override the Operating System settings from the **Cloudpacing Server**. When done this way, **Cloudpacing Player** will display a dialog warning the user.

The dialog provides a warning that testing indicates that the application works with the specified Windows OS though the vendor information does not specify the OS as compatible:

Important

*Microsoft has discontinued support for Windows Vista, Windows 8/8.1, Windows Server 2008 and Windows Server 2012. As a result, Numecent no longer supports use of these operating systems with **Cloudpaging Studio**. Windows XP, Windows 7, and Windows Server 2003 are still supported for legacy application packaging.*

- Windows 8.1 to 11 / 2012 R2 to 2022 (64-bit) – Check this box if the application supports running on a 64-bit version of Windows 8.1 to Windows 11 or Windows Server 2012 R2 to Windows Server 2022, since these platforms have the same requirements.
- Windows 7 (64-bit) – Check this box if the application supports running on 64-bit version of Window 7.
- Windows 8.1 to 10 (32-bit)– Check this box if the application supports running on a 32-bit version of Windows 8.1 or Windows 10. Windows 8.1 and Windows 10 have the same requirements.
- Windows 7 (32-bit) – Check this box if the application supports running on a 32-bit version of Window 7.
- Compatibility Settings – This button brings up a dialog that provides several compatibility options to be used when launching the cloudified application. For instance, this may be needed for applications designed for an operating system older than the target operating system(s). Which options are accessible depends on the selected target operating systems. For instance, if only Windows 7 is selected, then new compatibility settings introduced in Windows 10 or later will be disabled. For more information, see [About OS Compatibility Settings](#).

Important

Appsets that are created on a 64-bit platform, regardless of whether the application is 32 or 64 bit, can only be modified or paged to a 64-bit target OS. Appsets created on a 32-bit OS can be used on a 64-bit OS but may not always work depending on the application requirements.

Data Compression

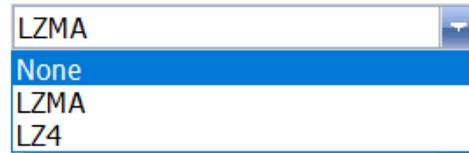
Compression is used to improve overall paging performance experienced by the end-user. The greater the compression ratio applied to data pages, the smaller and faster an appset will page. There are two-page compression settings to choose from:

- **None** – Indicates that no compression will be performed. This is useful for testing an appset but should never be used for paging.
- **LZMA** – (default) Uses LZMA compression techniques and is fast to decompress but takes longer to compress. This is the suggested compression method as it results in typically 20% smaller appset size than no compression, which results in a greater transfer rate when paging.
- **LZ4** – Uses LZ4 compression techniques and is faster to compress and decompress but may not result in sizes as small as LZMA. This compression is good for using with fast networks with low latency, such as with precache or multi-source.

NOTE

During the initial appset creation, optimization, and testing phases, it is most efficient to use “None” as the compression setting, in order to speed up the cloudifying process as much as possible. Then, after Optimization is completed, the compression can be re-set to the optimal level based on the particular application requirements and user needs.

Compression:



Data Encryption

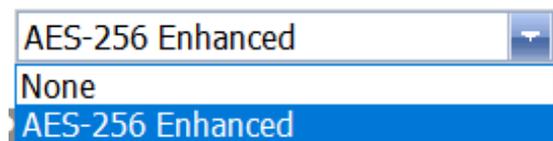
Encryption is used to help protect the appset as it is delivered and stored by **Cloudpaging Player**. The better the encryption method is applied to data pages, the harder it is to steal the content stored in that page of data and the more secure the application will be.

- **None:** Indicates that no encryption will be performed. This is useful for testing an appset but should never be used for real application delivery.
- **AES-256 Enhanced** (default): This is a Numecent enhancement of AES-256 that combines encryption and checksum into a single operation. It is utilizing the standard AES-256 encryption and decryption routines.

NOTE

Encryption methods will slightly degrade application launch time performance compared to no encryption due to the time needed to decrypt each page of data and verify its checksum. If this overhead is not desired and security is not a concern, then the encryption method can be reduced to “None” for improved application performance.

Encryption:

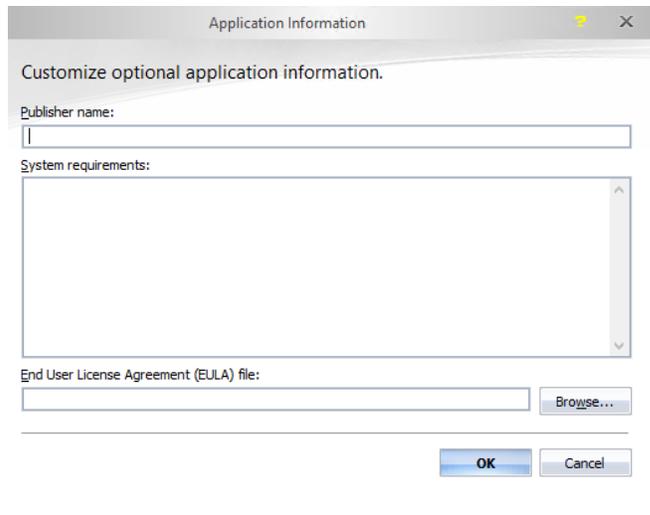


Other Application Settings

Additional, optional settings are accessible via the Application Information dialog.

To change the Application Information Settings

From the Project view, click the **More Information...** button. The Application Information dialog appears.



- **Publisher Name** – The name of the company that published the application or game, such as Microsoft or Electronic Arts.
- **System Requirements** – These are the hardware and software requirements provided by the manufacturer and are typically printed on the software packaging. These requirements are for informational purposes only and can also be viewed from the **Cloudpaging Server UI**.
- **Application End User License Agreement (EULA) file** – To populate this field, navigate to the EULA file. This is typically an HTML file that consists of a legal agreement the user acknowledges regarding the warranties and conditions of use for the software program.

Step 3: Start the Application Installation Capture Process

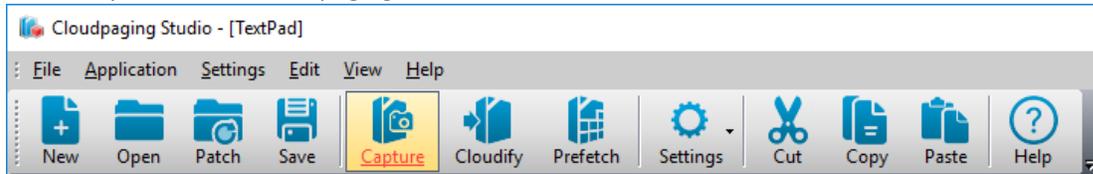
Before you start the capture process, it is recommended that you save the project. You should also become aware of where files and registry settings are stored. The more you know about this before the capture process, the easier it will be to optimize the appset later.

Before launching the capture process, you can select the default disposition for new files, folders, and registry keys via the Default Configuration settings.

For more information on disposition, see [Virtualization](#).

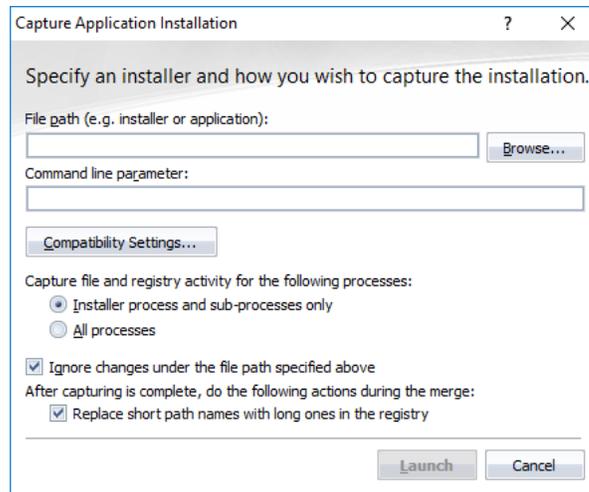
To start the capture process:

- 1) Click Capture on the Cloudpaging Studio toolbar.



Alternative: Click File on the menu bar, then select Capture Application.

The Capture Application Installation dialog appears.



- 2) Click **Browse** and navigate to the application's executable (**.exe**) or installer file (**.exe** or **.msi**) for the **Installer Path**. While this may typically be an installer, it could be the application's main executable depending on what needs to be captured.

NOTE

You should copy the installer under a temporary folder and launch it from there. This can help to avoid artifacts compared to launching it from a common location like the desktop.

- 3) If any arguments are required to run the installer, supply them on the **Command line parameter** field.
- 4) Under **Capture file and registry activity for**, select either **Installer process and sub-processes only** (default) or **All processes**. Then if required, select **Compatibility Settings**.
 - **Installer process and sub-processes only** – When this option is selected, Cloudpaging Studio will record only file and registry activity coming from the selected process and its subprocesses. All other system activity is ignored.
 - **All processes** – When this option is selected, file and registry activities coming from all processes on the system are recorded. Unwanted artifacts could result from capturing All Processes. For example, browsing the Internet while capturing the installation of an application may cause a change in the browser cache, which will also be captured.
 - **Compatibility Settings** – This dialog provides several compatibility options to be used when an installer is launched. For instance, enabling Compatibility Mode may be necessary for installers designed for an operating system older than the Cloudifying PC operating system.

Which options are accessible depends on the Cloudifying PC operating system. For instance, under Windows 7, new compatibility settings introduced in Windows 8 will be disabled. For more information, see *About Compatibility Settings* at the end of this chapter.

- **Ignore changes under the file path specified above** – This option is selected by default and is used when capturing an installer, or an external updater that is not part of the application itself, to prevent capturing temporary files that are created under the installer's folder during the installation process. Deselect this option when capturing the application itself, however, especially when performing updates from within the application, otherwise any new or updated files under the application's installation folder will be removed from the workspace after the capture completes.

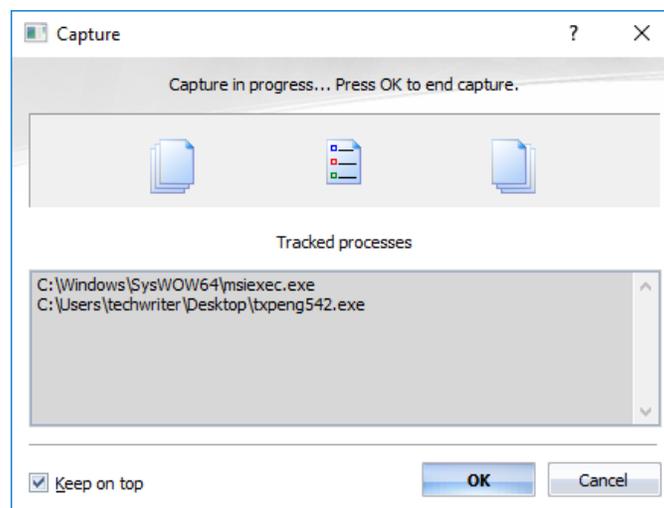
Important

If Ignore changes under the file path specified above is checked, installing an application to the same folder as the installer will cause files needed for proper capture to be excluded from the package. We recommend that the install source path and the install target path be separate folders.

- Replace short path names with long ones in the registry – Some applications still refer to their component files by their 8.3 DOS-style file and folder names. For these applications, uncheck this box.

5) Click **Launch**.

The Capture in Progress dialog appears and the application installer launches.



During the capture, the bottom half of the Capture in Progress dialog shows a list of every process captured during the installation.

NOTE

Leave this dialog open. Later, when the application installation process has completed and you want to end the capturing process, you will click OK.

Step 4: Install the Application

Follow the application installer instructions to complete the application installation. Be sure you install all components during the installation process to minimize any dependencies. Install the application to the Install Folder.

Most applications have Standard and Custom installation options. It is recommended that you use the **Custom** option so that all application components will be available to users of varying needs.

Installation Folder Restrictions

The folder where the application will be installed should not be the root directory of the hard drive. This will cause **Cloudpaging Studio** to filter nothing. Though technically this will not prevent proper application paging, the appset will contain unneeded data and be larger than required. Do not use special characters in the installation directory name (e.g. no umlauts, ~, !, @, #, à, Á etc.).

Important

Cloudpaging Studio will not capture installation to a network drive.

We recommend that the installed application should not be in the same folder as the installer executable or in a sub-folder of the installer folder.

We also recommend that you install to the default location, such as Program Files, but with a unique folder name. Be sure this folder name is unique as it will help to avoid issues with legacy application that still use the older 8.3 naming convention (short file names) that can cause integration conflicts.

Important

Cloudpaging Studio does not parameterize the Non-OS drives. If you install and capture an application on such a drive, the same drive letter must exist as a fixed drive on the client machine otherwise the appset will not be able to virtualize.

Avoiding Artifacts

Artifacts are unwanted changes or additions that occur most often when “All Processes” is selected on the Capture Application Installation dialog. Follow the guidelines below to avoid or reduce artifacts.

- Avoid running applications or executing commands that are not related to the target application on the cloudifying machine while capturing installation and other functions are taking place.
- Do a dry run of the installation on a machine other than the one used for cloudifying so that you know what to expect.
- In general, minimize all machine activity after any cloudifying activities begin.

Important

You should turn off any power saving features that would cause the system to go standby or hibernation during the capture process.

If you selected the “launch program after installation” option at the end of the application installation, continue to Step 6. If not, skip to Step 7.

Step 5: Run the application (optional)

The purpose for running the application at this point is to capture settings (such as registry modifications) that are made after the installation is completed. Such cases are processes such as displaying the EULA (End User License Agreement) or changing user profile settings. After the installation is complete, it is advised to run the application and try using some menu options and features to verify that related components are properly installed. Some suggested actions are:

- Start the application
- Click and use menu items
- Click and use Toolbar items
- Perform a typical task and usage scenario for the application. Refer to the user guide for the application.
- Close the application

NOTE

For applications that require running other processes after the installation is complete, you may need to stop the capture, then start it again for each process.

Step 6: Capturing System Reboots

If the installation requires a system reboot after the installation is complete, then take the following steps:

- 1) Click the **restart** button from the application's installer. Do not end capture or exit **Cloudpaging Studio**, as it will save work in progress and shutdown on its own.
- 2) Once the system restarts, login is required.
- 3) **Cloudpaging Studio** will startup automatically in capturing mode and will prompt you with a message box if there are additional Run key process that can be captured.
- 4) Select **Yes** to the additional processes that you wish to capture. You will be prompted for each run key contained in the registry.

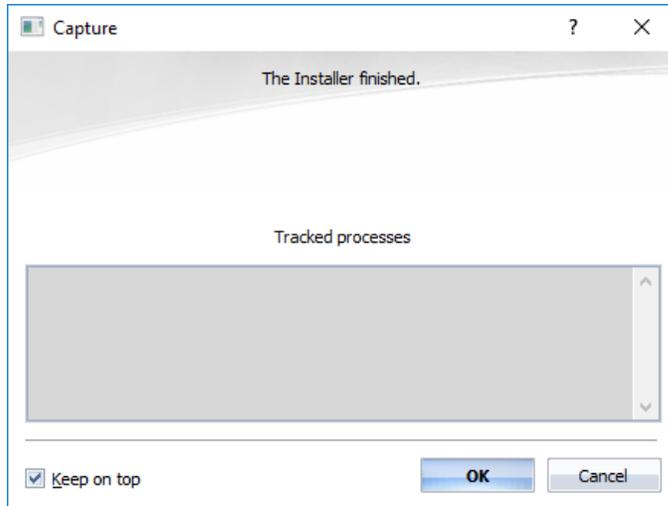
NOTE

There may be other process set to run that are not part of the application installation. Please be sure to select the correct processes.

Step 7: Stop the Installation Capture

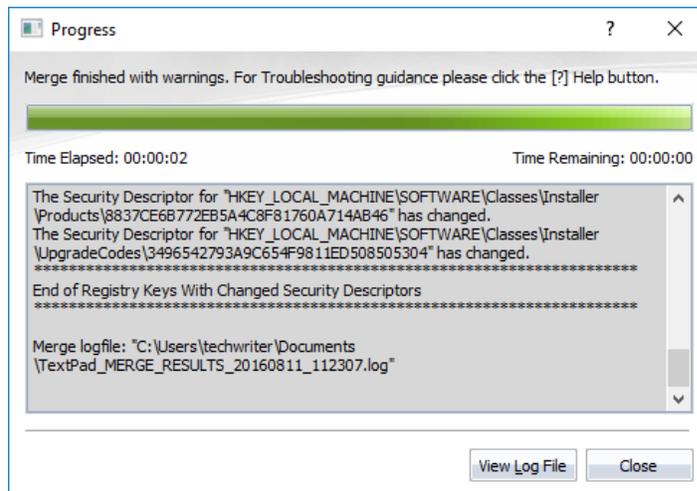
When the installation is complete, the Capture in Progress dialog will display a message that the capture is complete.

To end the capture process, click **OK**.



Be sure that the installation is complete before you stop the capture process. After you stop the capture process, the captured assets will be added (merged) into the project, and the project workspace will be updated to reflect these changes.

The Merge dialog will show which files and registry keys were filtered, merged, or had their security descriptions changed during the merge.



This information might be useful for troubleshooting. For example, if the installer changes the security settings for a file, you might need a Configurable AppEvent (CAE) to perform the same task on the Target PC.

Cloudpaging Studio also saves the merge information to the log file. To see this, click **View Log File**.

NOTE

Occasionally, you may see a message about non-critical errors during the merge of the captured data. These are due to protected objects by the Operating System not being accessed and can safely be ignored.

If you see an error message about security descriptors, User Accounts, or Plug-n-play drivers, you will need to create a Configurable AppEvent to handle these. See [Configurable AppEvents Settings](#) for more information.

There is no limit to the number of captures that you make. At any time you can re-configure project settings and take another capture.

Step 8: Enter Remaining Application Information

After capturing the installation application, there are a few more pieces of information that **Cloudpaging Studio** will require before the next step.

Revision Information

The Revision Information sets the name of the current patch version, command line, and working folder for the application.

Revision information

Title:
Original Version

Command line:
?programfilesx86?\TextPad 5\TextPad.exe

Working folder:
?programfilesx86?\TextPad 5

Revision number: 0 Last edited with: Cloudpaging Studio 9.1.0.20006
Format version: 21 Last captured on: Windows 10 64-Bit

- **Title** – This is the patch name of the current version of the appset. It defaults to “Original Version” the first time you capture an application. When you create a patch ([Chapter 9, Patching an Appset](#)), you can assign the patch a description name, such as the actual version number and/or the release date.
- **Command line** – Specifies the command line used to start the program. When the user clicks the **Launch** button on **Cloudpaging Player**, the executable in the Command Line executes. The easiest way to populate it properly is to navigate in the Files view to the application’s executable that is to be launched, right-click it, then select **Set as command line**. Once set, the file marked as the Command Line will appear in **bold** in the Files view.

If you’re not sure where the executable is located, one way it can be found is by looking at the properties of the shortcut created when the application was installed.

Important

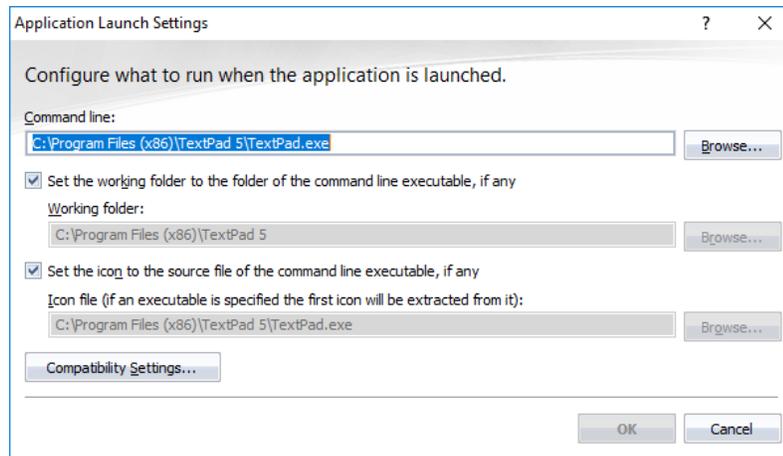
Setting the **command line** manually from the Project View will not set any security defaults for the file chosen. In addition, file paths containing templates (see [About Folders](#)) will need to be substituted manually.

- **Working folder** – Specifies the folder that contains the application’s executable or some related files. A vast majority of the time, using the same directory as used in the command line will work. On occasion using the same directory as where the program executable is found will not work. In these cases, it may be required to set the Working Directory to the application root directory. The easiest way to populate this field is to navigate in the Files view to the folder that you want to set as the working folder, right-click it, then select **Set As Working Folder**. The working folder path will be shown in **bold** in the Files view.
- **Revision Number** – If this appset has been patched, this number will show the number of patches that have been applied.

- **Format Version** — This number indicates which version of the Studio packager was used to create the current Appset. This information can be useful if you need to contact Numecent technical support.
- **Platform** — This indicates which version of the operating system was in use when the appset was created.
- **Notes** — Click this button to add or read any notes relating to this appset.

Setting the Command Line and Working Folder from the Project View

Click the **Launch Settings** button to view the Application Launch Settings dialog.



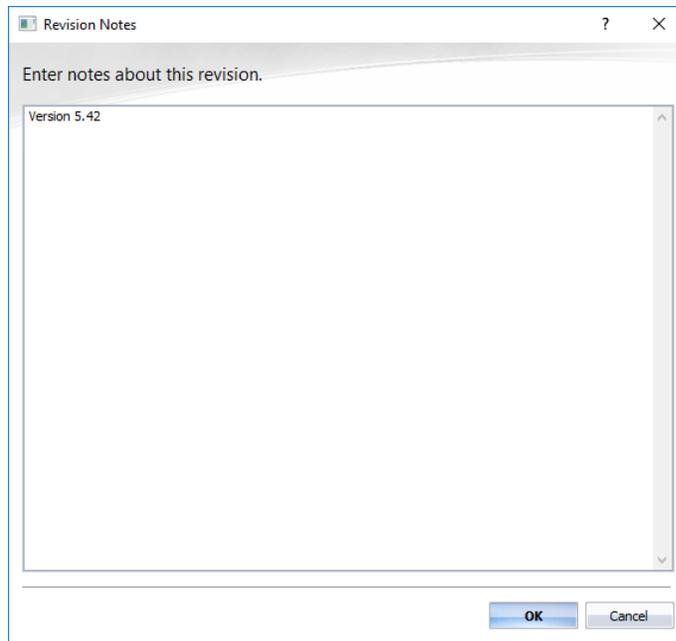
- To set the **Command Line**, type the path to the desired executable file or browse for it. If the file for the Command Line is a shortcut (*.lnk file), then the target of the shortcut will be used.
- To set the **Working Folder**, type the path to the desired folder or browse for it.
- The target file for the **Application icon** is automatically populated when an .exe file is set as the launch command. An icon can be set manually by specifying an .ico file or by extracting an icon from any .exe or .dll file.

NOTE

When extracting an icon from an .exe or .dll file, Cloudpaging Studio will use the first icon (Index 0) in that file.

Revision Notes

Revision Notes provide you with a space to enter information regarding each revision of the appset. To enter notes for the initial version, click the **Revision Notes** button. The **Revision Notes** dialog appears.



A new revision note is entered every time the appset is patched. Each revision note is saved as a separate text file in the appset file.

For information on patching, see [Chapter 9, Patching an Appset](#).

Step 9: Save the Project

Click **Save** on the **Cloudpaging Studio** toolbar to save the project (.stw) file.

When saving a project (.stw) file created with a version of **Cloudpaging Studio** prior to 9.0, **Cloudpaging Studio** will offer to update the workspace file to the new 9.0 format. This format is required to share and reuse the project file. If you choose not to update the file, **Cloudpaging Studio** will save the project file in the older format.

For more information on reusing project files, see [Reusing Project Files](#).

NOTE

When updating the project file to the new format, we suggest doing so on a computer as similar in configuration as the computer where you originally created the file. Of course, using the original computer would be the ideal situation.

Important

*Once you have updated a file to the new format, you cannot open it in versions of **Cloudpaging Studio** prior to 9.0. If you need to open a project in older versions of **Cloudpaging Studio**, you should maintain a backup copy.*

About OS Compatibility Settings

The **Compatibility Settings** dialog offers several options that affect the environment in which a program is launched. Those options are useful for programs designed to run on older operating systems than the ones on

which they will actually be run. For **Cloudpaging Studio**, this dialog appears under two different conditions: the program may be the installer for the application being cloudified or the cloudified application itself.

If you change compatibility settings for a given program, then Windows automatically modifies the runtime environment upon launching the program to make it as close as possible to the environment required for the program to function properly.

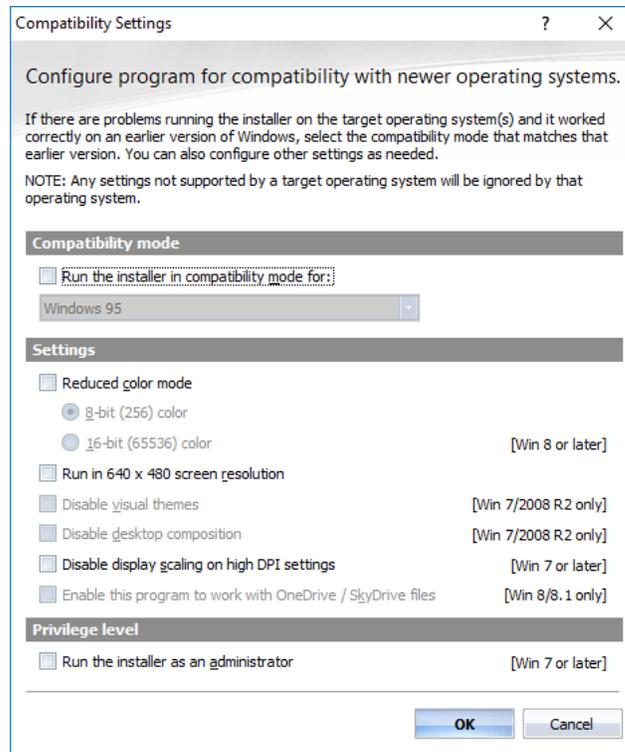
Compatibility Settings options

Certain compatibility settings apply only to some of the supported operating systems. This is clearly indicated next to each setting.

NOTE

Since multiple target operating systems may be selected, Cloudpaging Studio allows you to choose any of the options, even those that do not apply to some of the target operating systems you selected. However, any setting that does not apply to a particular OS will be ignored when the application is paged on that OS.

- **Run this program in Compatibility Mode** – This setting is used when a program is only compatible with an operating system older than the one on which it will run. Since **Cloudpaging Studio** only runs on Windows 7 or later, this option should be used for applications that only run properly on prior operating systems such as Windows 9x or Windows 2000. If the application indicates compatibility with Windows 7 or later, then this option does not need to be selected. Otherwise, as a rule of thumb, the most recent supported operating system should be selected for compatibility mode.
- **Reduced color mode** – This setting forces a program to run with a limited set of colors. Two options are available: **8-bit (256) color** and **16-bit (65536) color** (Windows 8.1 and later). This may be necessary for older programs, especially games, that do not support higher color ranges such as 24-bit or 32-bit color.
- **Run in 640x480 screen resolution** – This setting forces the entire screen to 640x480 resolution while the program is running. This may be necessary for older programs, especially games, that do not support higher resolutions. Try this setting if the graphical user interface appears jagged or is rendered improperly.
- **Disable visual themes** (Windows 7) – This setting turns off all visual themes while the program is running. Try this setting if you notice problems with the menus or buttons on the title bar of the program.
- **Disable desktop composition** (Windows 7 only) – This setting turns off transparency and other advanced display features. Choose this setting if window movement appears erratic or you notice other display problems.
- **Disable display scaling on high DPI settings** (Windows 7 and later) – This setting turns off automatic resizing of programs if large-scale font size is in use. Try this setting if large-scale fonts are interfering with the appearance of the program.
- **Enable this program to work with OneDrive/SkyDrive files** (Windows 8.1 only) – This setting allows a program to see and open files on SkyDrive that may not be otherwise visible.
- **Run this program as an administrator** (Windows 7 or later) – This setting causes Windows to elevate the program to administrator level before running it. Some programs require administrator privileges to run properly.



For more information on a particular setting, search one of the following web sites:

- <http://support.microsoft.com/>
- <https://technet.microsoft.com/>
- <https://msdn.microsoft.com/>

Chapter 4: Cloudifying the Application

As described in *Cloudifying Overview* chapter, application cloudifying with **Cloudpaging Studio** has four phases.

Phase 3: Cloudify the Application includes the steps necessary for creating an *appset*, or “cloudified application.” In addition, procedures for modifying the content and settings of files, registry, and environment are provided. These modification procedures also apply to Phase 5 (testing and optimizing the appset), as well as modifying a patched appset (Chapter 5).

The following are explained in this chapter:

- Procedure for Cloudifying the Application
 - Configure Cloudpaging Settings
 - Modify Files, Registry, and Other Assets
 - Cloudify the Application

Procedure for Cloudifying the Application

Step 1: Configure Cloudpaging Settings

Check the Settings dialog and modify any of the setting which affect the behavior of the cloudified app on the Target PC.

- [Configurable AppEvents Settings page 5](#)This configures the cloudified application to launch scripts at different stages of the paging process to perform specific actions on the Target PC as required by the application.
- [Virtualization Settings page 5](#)This controls how the cloudified application interacts with the Target PC.
- [Sandboxing Settings page 5](#) The cloudified is kept isolated from other programs and assets on the Target PC, but this can be modified or disabled for specific assets using these setting.
- [Security Settings page 60](#). This allows you override file security settings for a specific asset if required for the cloudified application to work on the Target PC.
- Multiple processes can be added to the security override list by repeating the steps. Each process can also be modified or removed as needed. [page 62](#)

This setting controls how much of the cloudified application can be stored locally by the **Cloudpaging Player** software.

NOTE

Before cloudifying, you should check the compression and encryption settings on the Project view to make sure those are set to the desired options. See [Data Compression](#) and [Data Encryption](#).

Step 2: Modify Files, Registry, and Other Assets

Noticing differences in Files, Registry and other assets before and after the capture process will provide a basis for what needs to be modified.

Important

Before cloudifying, you should check those views to see if there are any conflicts (items in red) and determine if these represent actual problems that could affect cloudpaging.

- [Modifying Files and Folders](#) Add or remove files and folders from the project or change their disposition.
- [Modifying Registry Entries](#) Add or remove registry keys and values, and change existing values.
- [Modifying Environment Variables](#) Add or remove environment variables.
- [Modifying Fonts](#) Add or remove fonts.
- [Modifying Startup Items](#) Add or remove startup items, including registry keys and Startup Menu items.
- [Modifying Services and Drivers](#) Add or remove Windows services and software drivers.
- [Modifying Shortcuts](#) Add or remove Windows shortcuts.

Step 3: Cloudify the Application

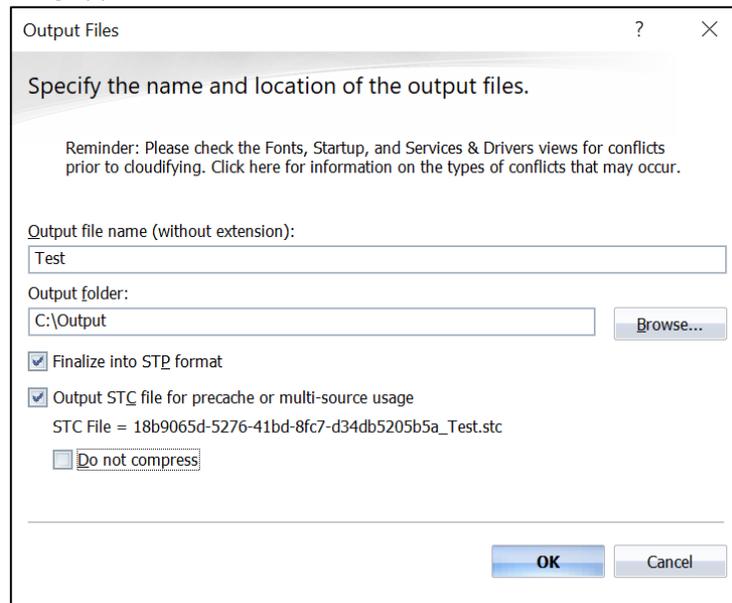
After you provide all information and configuration settings in the Project, File, Registry, and Environment views, you are ready to create the appset.

To cloudify the target application:

- 1) Click Cloudify on the Cloudpaging Studio toolbar.



- 2) The **Output Files** dialog appears:



This dialog lets specify a name and an output folder for the final appset.

By default, **Finalize into STP format** will be checked. **Cloudpaging Studio** will compress the components into the containing STP file. If the box is unchecked, the resulting STC file (the code and other configuration information that comprises most of the application), AIB file (metadata that describes the content of the STC file), TOK file (comprised of security data), and any other files such as icons, AppEvents, and EULA's, will be saved to the output folder.

To improve performance, it may be desired to have application pages located in a location closer to the Target PC. **Output STC file for precache or multi-source usage** makes it possible to have multiple locations for the application pages beyond that of a Cloudpaging Server. The location can be a company network drive, for example. Sources are paged from in sequential order of Local/ Remote Drive, and Cloudpaging Server. It is the responsibility of the administrator to place the output STC in the appropriate location. It is recommended to **Do not compress** the output STC when the desired use will be for precache.

NOTE

For instructions on how to manually finalize the cloudified app into the STP format, see [Saving an Appset File](#).

Saving the components without compressing them is much faster than creating the STP file. This allows you to check and edit the resulting AIB and TOK files, before creating the finished Appset.

Important

*Periods are not allowed in the appset filename. If you manually rename the appset, be sure NOT to use more than one period in the filename. Using multiple periods will prevent **Cloudpaging Server** from adding the appset.*

When the appset is later *published* to the **Cloudpaging Server**, the target application (or appset) becomes available for paging to users.

Chapter 5: Testing and Optimizing the Appset

As described in the *Cloudifying Overview* chapter, application cloudifying with **Cloudpaging Studio** has four phases. Phase 4: Testing and Optimizing the Appset includes the steps necessary for testing, optimizing, and re-testing the appset each time a setting changes. When the appset reaches optimal performance, the cloudifying process is complete and the appset can be published to **Cloudpaging Server**.

The following are explained in this chapter:

- Testing the Appset
 - Test the appset
 - Identify and Solve Problems
- Optimizing the Appset
 - Create Prefetch Files
 - Set Compression
 - Set Encryption
 - Cloudify the Application
 - Optional: Save the Cloudifying Environment
 - Publish the Appset to Cloudpaging Server

Testing the Appset

Step 1: Test the appset

To test the appset, you will load it onto a Target PC with **Cloudpaging Player** installed, then run the application. *Local paging*, the process of paging an appset to **Cloudpaging Player** from a file located the Target PC, allows you to test several appset iterations quickly without publishing the appset to **Cloudpaging Server**.

It is necessary to go through as many testing scenarios as are required, for different operating systems, application versions, etc. In this case, a clean version of the Cloudifying PC needs to be made and a new appset created for each different installation.

The Target PC can be one machine or multiple machines. Multiple PC's would allow testing to occur in parallel. The only requirement for the Target PC is that a clean installation of the operating system be present. This is mainly because at the end of the installation, the application resides entirely on the Cloudifying PC and testing on a clean machine can help identify missing components.

Important

Be sure not to use the Cloudifying PC also as the Target PC to test application paging.

NOTE

Local paging an appset can only be used for 30-days after the application is packaged. The exact date can be found when editing the TOK file by viewing the Application Expiration Time under the Application Times folder. Please use Cloudpaging Server to test after 30-days.

To test the appset:

- 1) Place [AppsetName].stp on the Target PC with **Cloudpaging Player** installed.
- 2) Rename the .stp file to [AppsetName].zip, and then right-click it and select **Extract Here**.
- 3) Double-click the .tok file. It will activate (outside of **Cloudpaging Player**).
- 4) Click **Launch** on the **Cloudpaging Player** toolbar. The target application will run.
- 5) At this point you can test the appset.

For example, all menu options should be tested to verify that all parts of the application are installed and captured. If the application does not work as expected, then some part of the target application installation was not completed properly.

Verify that the default location where the application saves files to is already part of the Excluded Paths for sandboxing rules (see [Sandboxing Settings](#)). This will ensure that user files remain on the system if the virtual application is removed.

NOTE

Optimizing an appset is typically an iterative process that requires testing between each successive cloudifying trial, as well as after each change in files, registry, or environment.

Step 2: Identify and Solve Problems

If you encounter problems or unwanted *artifacts* (dlls, etc.) when you test the appset, follow this procedure for identifying and solving problems:

- 1) First, identify the problems.
- 2) Second, modify the appset's files, registry, or environment to correct the problems.

Optimizing the Appset

After you have corrected existing problems, it is best to tune the performance of the appset. Delivery of applications can be optimized in two ways: by optimizing paging and/or by caching. To assist with these optimizations, please follow these basic steps.

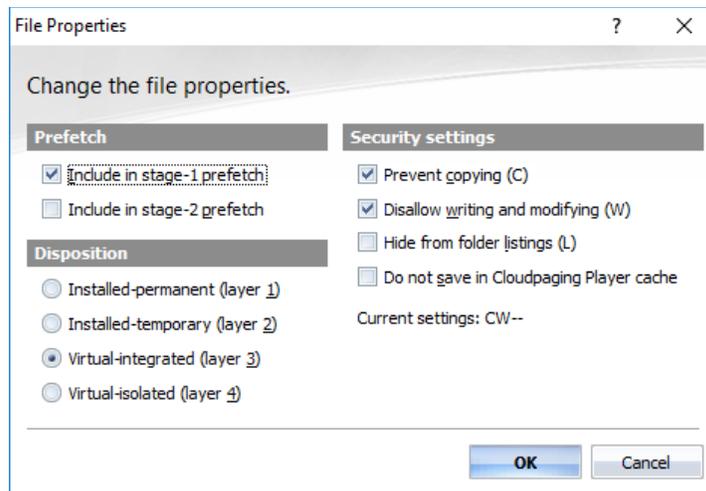
Step 1: Create Prefetch Files

One of the best ways to optimize the performance of the appset is to create a Stage-1 prefetch, and possibly a Stage-2 prefetch file. *Prefetching* is the process of fetching application pages prior to their being explicitly requested by **Cloudpaging Player**, in anticipation of being needed by the application. For more information, see [Prefetching](#).

- **Stage-1 Prefetch** – Application pages needed to start the application. Always recommended for best performance to launch an application.
- **Stage-2 Prefetch** – Progressive background fetching of additional application pages after application starts, improving overall performance.

To manually add a file to the prefetch:

- 1) View the **Files** tab of the **Cloudpaging Studio** workspace.
- 2) Select the file to be added to the prefetch and right-click to select **File Properties**.
- 3) Check the Prefetch box for adding to either Include in stage-1 prefetch or Include in stage-2 prefetch.
- 4) Click **OK**.

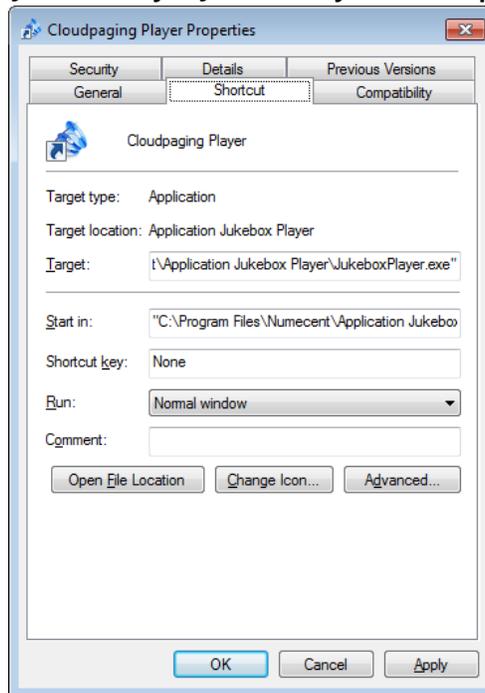


To capture and add a prefetch:

- 1) Enable the prefetch **Capture** button on **Cloudpaging Player**: right-click the **Cloudpaging Player** shortcut on the Windows desktop, and select **Properties**.

The **Cloudpaging Player Shortcut Properties** dialog appears.

- 2) On Target, append **/CapturePrefetch** to the path as follows: "**C:\Program Files\Numecent\Application Jukebox Player\JukeboxPlayer.exe**" **/CapturePrefetch**.

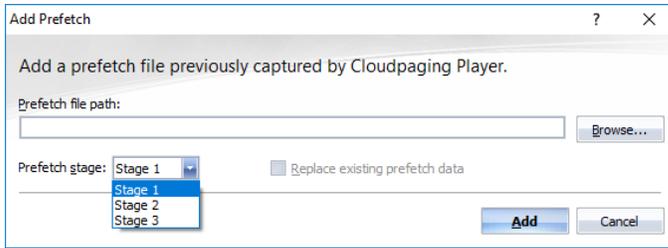


- 3) Click **OK**, and launch the **Cloudpating Player**.



The **Cloudpating Player** launches with the Prefetch **Capture** button enabled.

- 4) Add the appset to the **Cloudpating Player** by paging it either locally or from a **Cloudpating Server**.
- 5) Clear the Player cache. (Click **File**, select **Options**, click **Clear Cache**, click **OK**.)
- 6) To start prefetch capturing, select the **Stage** and click **Capture** on the **Cloudpating Player** toolbar.
- 7) A save dialog will appear for the filename to store the captured information. By default this will save to **My Documents** as <application's short name>_<application ID>_Stage_<stage #>.pf.
- 8) From the target application, perform the actions to include in this stage of prefetch, then click **Capture** to end the capture process.
- 9) Click the **Capture** button on the **Cloudpating Player** toolbar to stop capturing.
- 10) If you wish to add another stage prefetch, repeat steps 4 - 9 by selecting different **Stage** numbers.
- 11) Open the **Cloudpating Studio** project file for that application.
- 12) From the toolbar, click the **Prefetch** button.
- 13) The Add Prefetch dialog appears.



- 14) Enter the prefetch file that was just captured by **Cloudpaging Player** into the **Prefetch file path** field.
- 15) Select the **Prefetch stage** for which the prefetch file will be used.

NOTE

If one or two prefetch stages are in use, three prefetch stages will be shown in the dropdown list. Applying the new prefetch data to stage three will create a third preset stage. The next time the Add Prefetch dialog appears, four preset stages will be shown. For more information on how Cloudpaging Studio uses prefetch stages, see [Prefetching](#).

- 16) Check **Replace existing prefetch data** if this will replace an older prefetch file previously added.
- 17) Click **Add** and repeat the process for other stages.

NOTE

If there is an existing prefetch stage, then the “replace existing prefetch data” check box will be enabled. This does not apply to files manually added to the prefetch.

Best Practices for Capturing Prefetch Files

When capturing prefetch files, you should repeat the same activities you want to capture. For example, if you want to capture the launching part of an application into prefetch:

- 1) Turn on capturing.
- 2) Launch the application.
- 3) Stop the application, and launch it again.
- 4) Repeat while watching Cloudpaging tray icon.
- 5) Stop capturing when you no longer see Cloudpaging tray icon spinning as you launch the application.

NOTE

Many applications do not always read the same set of blocks every time you launch. An application will capture many of the same blocks during every launched but there are always some extra blocks that the applications will need at each launch, depending on circumstances. It normally takes 3 to 4 launches for an application (especially large applications) to capture most of the blocks usually requested during launch.

While capturing prefetch files on the Player:

- 1) Allow all prefetching to complete.
- 2) Stop other applications that might be running.

- 3) Clear the Player cache.
- 4) Start a prefetch capture for the application.
- 5) Launch the application.

Step 2: Set Compression

During the testing phase, compression is usually set to “None” in order to speed up the cloudifying process. After testing is complete, you can change the compression level to **LZMA**, which is the recommended setting as it typically yields the best compression ratio. This will result in faster download times when paging the application. Be prepared that this can take some time to cloudify the application.

NOTE

See [Data Compression](#) for more information on what compression method to choose.

To adjust the compression:

- 1) Open the **Cloudpaging Studio** project file for the application.
- 2) Select the appropriate compression level under **Data compression**.

Step 3: Set Encryption

Typically, encryption should be set to the default method **AES-256 Enhanced** to ensure a higher level of security for the application content and better performance.

NOTE

See [Data Encryption](#) for more information on what encryption method to choose.

To adjust the encryption:

- 1) Open the **Cloudpaging Studio** project file for the application.
- 2) Select the appropriate encryption method level under **Data encryption**.

Step 4: Cloudify the Application

After optimizing the application, you need to re-cloudify the appset.

To cloudify the application:

- 1) Click **Save** on the **Cloudpaging Studio** toolbar to save the project file (**.stw**).
- 2) Click **Cloudify** on the **Cloudpaging Studio** toolbar.

Step 5: Optional: Save the Cloudifying Environment

Patches can be created on any PC, as long as the most recent version of the appset has been extracted on that PC. If you wish to use the original Cloudifying PC for patching, follow these steps: Save the project file (**.stw**), appset file (**.stp**), and any prefetch capture files (**.pf**). Or back up the cloudifying environment.

Important

The original application must be installed in order for a patch to be properly integrated. This must either be done on the original Cloudifying PC or the appset can be extracted to another PC. The original project file (.stw) is also required.

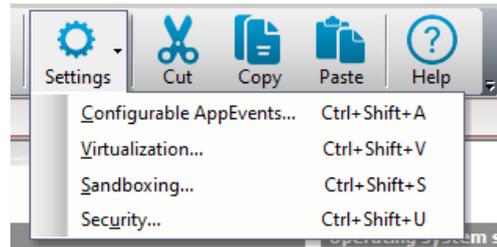
Step 6: Publish the Appset to Cloudpaging Server

Once optimized, the appset is ready to be *published* from **Cloudpaging Studio** to the **Cloudpaging Server**, which is done by copying the final STP file onto the **Cloudpaging Server** repository. The application is then ready to be paged to users' PCs that are running **Cloudpaging Player**.

Chapter 6: Cloudpaging Settings

Cloudpaging settings control the behavior of the cloudified application when it is paged or virtualized on the Target PC.

They are accessed via the **Settings** button or the **Settings** menu, which displays the **Settings** dialog.



To access the Settings dialog, either click the Settings menu and select Show Settings, or click the Settings button on the toolbar.

This dialog has several separate views, which are described in the corresponding sections below:

Configurable AppEvents, Virtualization, Sandboxing, and Security.

To access a specific screen on the Settings dialog, select that screen name from the Settings menu. You can also click the arrow next to the settings button and select the screen name from the drop-down menu.

The following are explained in this chapter:

- Configurable AppEvents Settings
- Virtualization Settings
- Sandboxing Settings
- Security Settings

Configurable AppEvents Settings

To convert a locally installed application into a cloudified application, **Cloudpaging Studio** captures the “native” states of the installed application and records them in the appset file. When an application is activated by the user, those states are re-created on the user’s system. Sometimes, to allow the application to run properly, it may be necessary to adjust those native states or the runtime environment before, during, or after activation. In other words, it may be necessary to perform certain actions that cannot be captured in an appset.

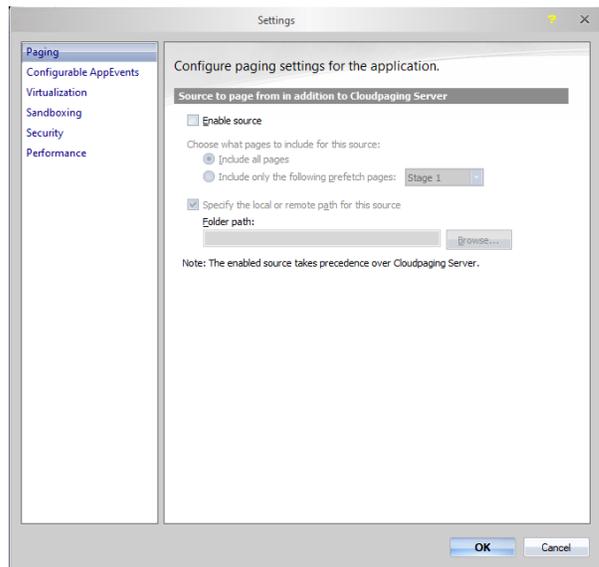
An *AppEvent* is an action that uses a *trigger* to run a *handler* (e.g., an executable program or script) that will perform specified runtime action(s) that are required by the application being cloudified, such as checking for prerequisites. AppEvent handlers can be written in any scripting or programming language, such as powershell, vbscript, or executable programs.

Examples of actions that may need to be performed at runtime are:

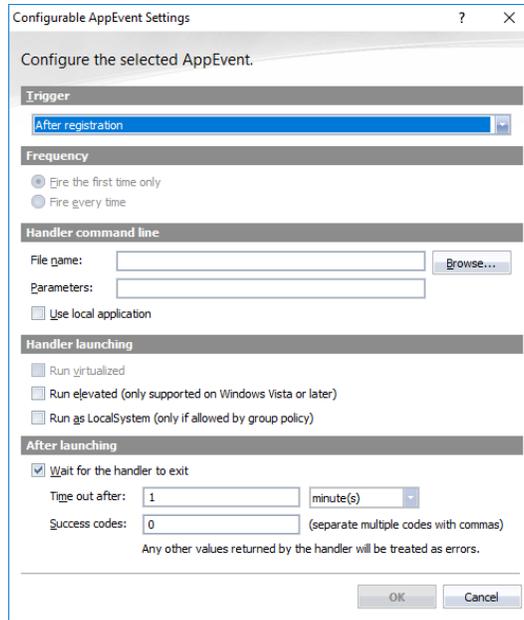
- Configuring plug-ins for office suites and browser applications
- Configuring virtual printers
- Configuring game parameters for a video game
- Starting a background program

To add an AppEvent

- 1) Select **Configurable AppEvent** from the Settings menu. The Settings dialog appears with the **Configurable AppEvents** screen visible.



- 2) Click the Add... button. The Configurable AppEvent Settings dialog appears



- 3) Select a **Trigger** from the drop-down menu. A *trigger* is condition that causes an *handler* to start, such as running a VB script.

AppEvent Trigger	Description
After registration	Occurs once after Cloudpaging Player checks OS requirements for the application.
After activation	Occurs once immediately after Cloudpaging Player downloads the AIB and prefetch files.
After virtualization	Occurs immediately after the player has virtualized all files, registry keys, environment variables, fonts, services, and drivers.
After launch	Occurs immediately after the application launches
After exit	Occurs after the application exits
Before devirtualization	Occurs before Cloudpaging Player removes all virtualized files, registry keys, environment variables, fonts, services, and drivers.
Before deactivation	Occurs once immediately before Cloudpaging Player removes all virtualized assets.

Triggers occurring after a specific state fire before **Cloudpaging Player** moves on to the next state. For instance, **Cloudpaging Player** will fire all “After registration” AppEvents before proceeding to activation, unless **Wait for handler to exit** is not checked.

- 4) Select the **Frequency** setting from the drop-down menu. *Frequency* is how many times an action is set to occur after the AppEvent is triggered.

Frequency	Description
Fire Once	The action will run the first time the AppEvent is triggered, and never again.
Fire Always	The action will be triggered each time the AppEvent occurs. Some triggers only occur once and this option will be disabled.

- 5) Select a **Handler command line** by clicking **Browse** to locate the name of the executable or script file, then select it.

The file name is displayed in the **File Name** field.

NOTE

See [About AppEvent Scripts and Executables](#) for important information about AppEvent scripts and executables.

- 6) Enter any necessary **Parameters**, which may include configuration switches or numeric values. For example: "visible = true" or "16" or "/faststart."
- 7) Select **Use local application** if the Handler executable or script is located on the local PC or on the company network. If this option is selected, the handler is not included with the appset (.stp file).

Important

Use local application can be used to specify an executable within the appset. However, there are some limitations. The triggers can only be "After virtualization", "After launch", "After exit", or "Before devirtualization". The handler will always be launched virtualized (regardless of "Run virtualized" setting).

- 8) Choose how the handler is to be launched:
 - **Run virtualized** - This option specifies if the handler should see isolated assets (i.e. files and registry keys) or not. It can only be specified for triggers that occur after an application has been virtualized.
 - **Run elevated** - This option specifies if the handler should be launched elevated requiring administrator privileges. It only affects handlers that execute on Windows 7 or later.
 - **Run as LocalSystem** - This option specifies that the handler should be launched as a SYSTEM process. This means the handler has access to files that the current user also does. Handlers run as LocalSystem can affect critical files and should be used cautiously. This option will work only if you have configured the Group Policy for **Cloudpaging Studio** to allow it.
- 9) Select the **After launching** options:
 - **Wait for handler to exit** - Requires **Cloudpaging Player** to wait for the handler to finish before continuing to the next state. If not set then the handler is executed in parallel with **Cloudpaging Player** continuing normal operations.
 - **Time out after** - Specify the amount of time, in seconds or minutes, **Cloudpaging Player** should wait for the handler to exit. If the handler has not returned by this amount of time, then the **Cloudpaging Player** will treat this as a failed return code.

- **Success codes** - The handler must generate a specific *return value* to **Cloudpaging Player** that indicates either success or failure. By default, 0 is the only success code and all other values are interpreted as a failure. However, if 0 is not the desired then new success codes can be specified here.

After launching options do not apply to the “post-launch” trigger as the application has already launched and cannot be interrupted.

Important

Existing appset will use the default success codes of 0 for success and all other return values are interpreted as a failure.

- 10) Click **OK**. The AppEvent has now been added.

About AppEvent Scripts and Executables

An AppEvent handler is a script or executable program, which can be written in any scripting or programming language, such as Visual Basic, C++, etc.

NOTE

There is a 10 MB size limit for each script. Also, scripts cannot be modified after they are cloudified into an STP file. Any modification will be seen as tampering and the script will be disabled.

Upon detecting a failed success code returned from the AppEvent handler, **Cloudpaging Player** will log a failure with the return code but will not display any messages for error codes. Therefore, when creating a script that will return an error code upon failure, it is necessary that the script itself displays any desired message to the user. If the failure is during the activation sequence, then the application will not be added to **Cloudpaging Player**. If the failure is during the removal process, then the removal will simply continue.

NOTE

During a Windows logoff or shutdown, AppEvent triggers will be skipped because Windows does not allow new process to be created during this time. Several sample Visual Basic AppEvent scripts are included with Cloudpaging Studio. These sample scripts check the local system for the existence of the applications shown in the table below.

Script File Name	Description of AppEvent Script	Error Code
AddService.exe	Used to add and start a PNP service	-1
RemoveService.exe	Used to stop and remove a service	-1
DetectSoftware.ps1	Checks is software is installed under Programs	1
DotNetCheck.ps1	Checks for existence of .NET Framework	1

OpenFirewall.ps1	Open firewall for programs or ports	-1
OSMemCheck.ps1	Verifies there is enough memory	1

NOTE

When using the sample PowerShell scripts included with Cloudpaging Studio, make sure to enter the Parameters setting in the Configurable AppEvent Settings dialog with the major and minor numbers of the required software version. For example, to detect .NET Framework version 4 use a Parameters setting of Optionally a title can also be set for the failure dialog (e.g., DotNetCheck.ps1 4 Framework)

About AppEvent Context

An AppEvent has access to environment variables that define specific information for the application being paged. An AppEvent can query this information to be used as needed.

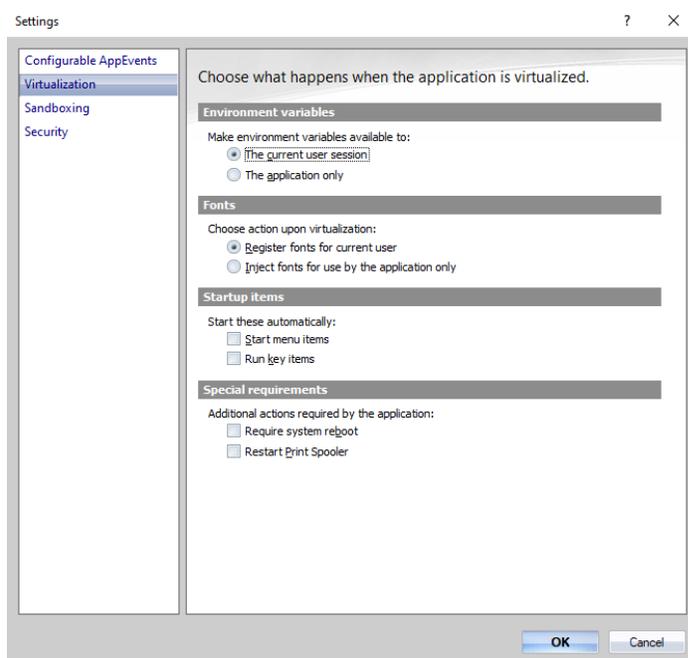
- **AppName** - The full name of the application
- **AppID** - The Application ID for this appset.
- **WorkingDir** - The **Working folder** set under the advanced section below.
- **CommandLine** - The **Command line** set under the advanced section below.
- **ClientId** - The unique Player ID used by the **Cloudpaging Server** to associate with a user names.
- **LastAccessTime** - The last time this application was launched if set.
- **ProviderUrl** - URL to the **Cloudpaging Server** or SaaS provider portal if set.
- **SessionKey** - Option token parameter set by a SaaS portal.
- **Extra** - User defined field that can pass information to the AppEvent. For more details, please visit <https://numecent.freshdesk.com/a/solutions/articles/1000302763>.

Important

When using the Extra parameter, be sure to not include the <, >, |, &, ^, ", #, %, {, }, \, ~, [,], ', or space characters unless they are encoded by the portal and decoded by the handler. All characters will appear as part of the value in the extra environment variable and do not need to be escaped.

Virtualization Settings

The Virtualization setting controls how the cloudified application interacts with the Target PC running **Cloudpaging Player** and whether specific parts of the application are visible to the whole system or just the cloudified application.



To access **Virtualization Settings** dialog, click the **Settings** menu and select **Virtualization**. The Settings dialog appears with the **Virtualization** screen visible.

- **Environment Variables** – There are two options. You can make environment variables in the appset available to:
 - **The current user session** – Select this option if you want the environment variables provided with the appset to be shared with all applications in the user’s session.
 - **The application only** – Select this option if you want the environment variables provided with the appset to only be used by the cloudified application. Other applications will not be able to access these environment variables.
- **Fonts** – There are two options:
 - **Register fonts for current user**– Select this option if you want the fonts provided with the appset to be shared with all applications in the user’s session.
 - **Inject fonts for use by the application only** – Select this option if you want the fonts provided with the appset to only be used by the cloudified application. Other applications will not be able to access these fonts.

NOTE

When selecting Inject fonts..., you should also set the disposition of the SystemFonts folder to virtual-isolated (layer-4).

- **Startup Items** – There are two options: **Start menu items** – Select this option if there are Startup items for the application under the Start menu that should be launched after virtualization. **Run key items** – Select this option if there are Run registry key items for the application under HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run that should be launched after virtualization.
- **Special requirements** – There are two options:

- **Require system reboot:** Select this option if you want the user to reboot their PC after virtualization. This is only needed if the application requires a reboot in order to run properly, such as to setup services.
- **Restart print spooler:** Select this option if there is a printer driver that needs to be setup after virtualization by restarting the printer spooler service.

Sandboxing Settings

The act of *sandboxing* means to separate changes made to the file system or registry from the real system on the Target-PC. **Cloudpaging Studio** performs partial-sandboxing for files, which means that only content modified or added to folders or sub-folders of root folders in the project are sandboxed. For example, if a project contains only the installation root folder and “Program Files,” and if the cloudified application creates a new file under the “Windows” folder, then this new file will be directly written to the system. **Cloudpaging Studio** will sandbox all default registry hives (e.g., HKCR, HKCU, HKLM, HKU), but any other hive must be part of the appset to be sandboxed.

For the paths in the project that are sandboxed, the content is separated from the system, meaning that they are not written to the local system but stored separately and only made available to the paged application. Again, using the example above, if a new file under “Program Files” is created, then it will be sandboxed and not written to the system. Only existing files under “Program Files” that are not contained in the appset would be modified by the application if attempted. The same rule applies for existing registry keys not contained within the appset. Sandboxing is not intended to be applied to documents or data saved by the user using the paged application. To help avoid this, data written to the following folders are the only exception for sandboxed folders:

Sandbox Exclusion Folder	Default Path - Windows 7
System Root	C:\
Desktop	C:\Users\ <username>\Desktop</username>
Documents	C:\Users\ <username>\Documents</username>
Downloads	C:\Users\ <username>\Downloads</username>
Music	C:\Users\ <username>\Music</username>
Pictures	C:\Users\ <username>\Pictures</username>
Videos	C:\Users\ <username>\Videos</username>
Links	C:\Users\ <username>\Links</username>
Favorites	C:\Users\ <username>\Favorites</username>
Contracts	C:\Users\ <username>\Contacts</username>
Saved Games	C:\Users\ <username>\Saved Games</username>
Searches	C:\Users\ <username>\Searches</username>
Temp	C:\Users\ <username>\AppData\Local\temp</username>

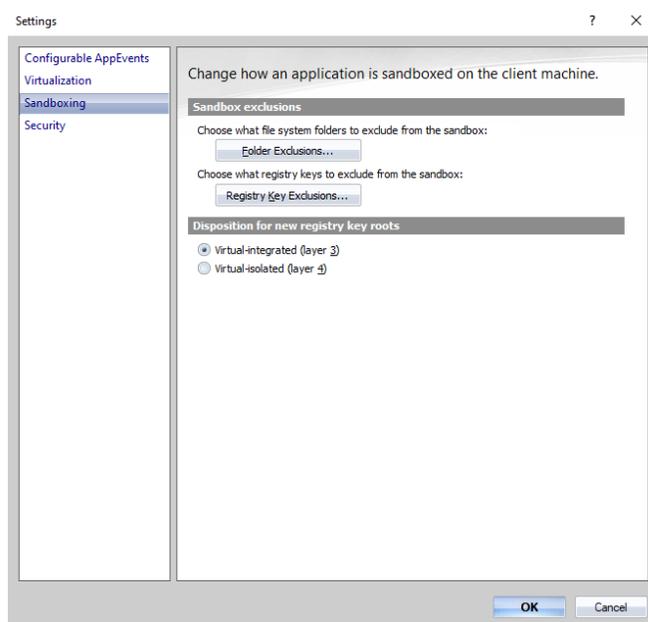
Internet Cache	C:\Users\ <username>\AppData\Local\Microsoft\Windows\Temporary Internet Files</username>
Shared Documents	C:\Users\Public\Documents
Shared Downloads	C:\Users\Public\Downloads
Shared Music	C:\Users\Public\Music
Shared Pictures	C:\Users\Public\Pictures
Shared Video	C:\Users\Public\Videos

ADD TABLE

Data written to those paths is

directly written to the system, even if the “Users” folder is part of the appset as a root folder, as would be expected by end-users. All the sandboxed content is deleted when the application is removed.

To access the **Sandboxing** settings, click the **Settings** menu and select **Sandboxing**. The Settings dialog appears with the **Sandboxing** screen visible.



Sandbox Exclusions – If necessary, specific registry keys and folders can be excluded from the sandbox. This may be required if the asset in question is shared between a cloudified application and another locally installed application.

To add a folder exclusion

- 1) Click Folder Exclusions.
- 2) The **Sandbox Folder Exclusions** dialog appears.
- 3) If the excluded folder exists on the Cloudifying PC, you can drag the folder from Windows Explorer into the **Sandbox Folder Exclusions** dialog.
- 4) If the excluded folder does not exist on the Cloudifying PC, click **Add...**
- 5) A blank line will appear on the **Sandbox Folder Exclusions** dialog. Type the full folder path and hit **Enter**.

To add a registry key exclusion

- 1) Click **Registry Key Exclusions**.
- 2) The **Sandbox Registry Key Exclusions** dialog appears.
- 3) Click **Add...**
- 4) A blank line will appear on the **Sandbox Registry Key Exclusions** dialog. Type the path and hit **Enter**.
 - **Default Disposition for new registry key roots** – Registry key roots created dynamically by the virtualized application while running will use the selected disposition:
 - **Virtual-integrated (layer 3)**
 - **Virtual-isolated (layer 4)**

NOTE

This setting replaces part of the Virtualization Settings option in previous versions of Cloudpaging Studio. The rest of this functionality is found under the Global Disposition Settings dialog. See [Globally Changing Asset Dispositions](#) for more information.

Security Settings

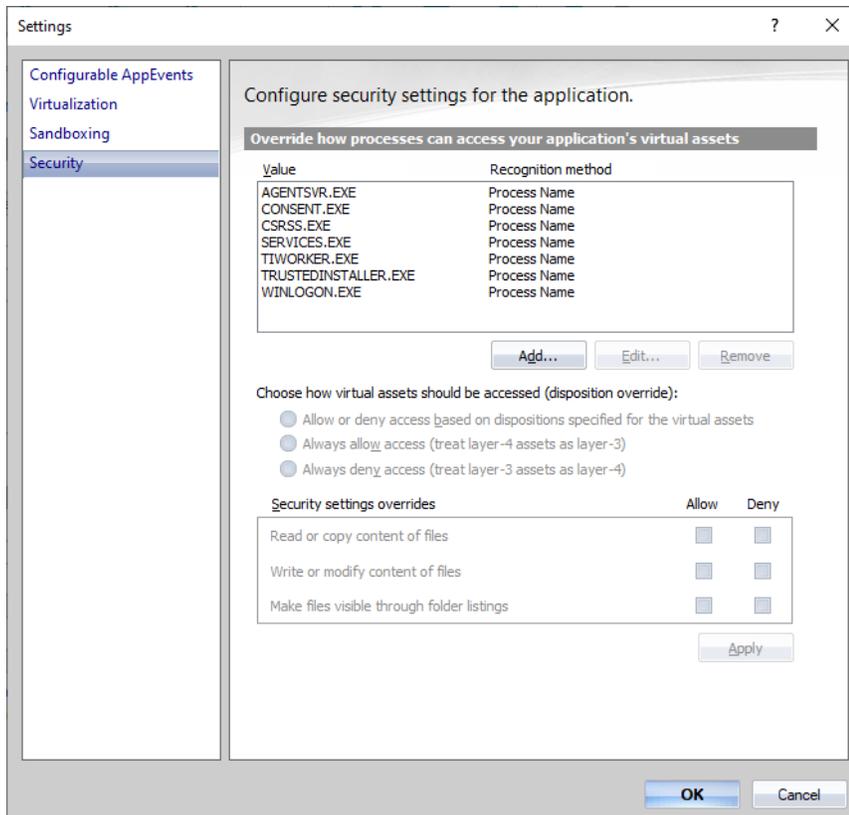
There are times when the standard file security settings (see [Editing Folder Properties](#)) need to be bypassed by specific application process. For example, an application patch installer may be required to modify a file that is protected with the “Make read-only” flag. In this case, a *security override* can be used to allow this to happen while still protecting the application. Another example would be to not allow antivirus or indexing software from fetching all the application pages. This undesirable side-effect can also be prevented by denying those specific processes from accessing the virtual application.

NOTE

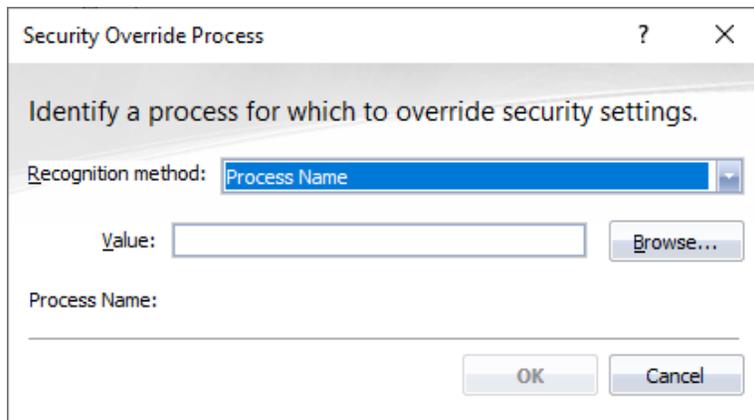
Security overrides only affect the file security settings but do not change the visibility of the disposition layer. If a file is layer-4, then setting a system process to override all the security settings will still not allow that process to see the isolated layer-4 file.

To set up a security override:

- 1) To access the **Security** settings, click the **Settings** menu and select **Security**. The Settings dialog appears with the **Security** screen visible.



- 2) Click **Add** to add a process name to the security override list. This will open the Security Override Process dialog.



- 3) Set the following information and click **OK** when complete:
- Recognition method** – There can be duplicate process names in the list that correspond to distinct file versions (e.g., multiple patch executables all with the same name). In order to distinguish between them, a recognition method must be used to authentic the process. The following types are supported:
 - Process name - Name of the process as it appears in Task manager (weakest method), that only verifies the process name as specified by the **Value** field.
 - Process path - Full path and name where the executable file is located on the local system, that will verify both the process name and file path as specified by the **Value** field.

- Service Name - Name of a specific service from as it appears in the Service manager, that will be used to verify the process by the service name as specified by the **Value** field.
 - **Value** – The process name, process path, or service name value based on the **Recognition** method selected.
- 4) Select whether to override the disposition settings for virtual assets.
- Allow or deny access based on the dispositions specified for the virtual assets – This setting causes the selected process to respect the individual disposition settings for files, folders, and registry keys/values.
 - Always allow access – This setting causes the selected process to treat all virtual asset as if they were virtual-integrated (layer 3), even if they are set at virtual-isolated (layer 4).
 - Always deny access – This setting causes the selected process to treat all virtual asset as if they were virtual-isolated (layer 4), even if they are set at virtual-integrated (layer 3)
- 5) Next set the process Allow and Deny access control and choose from the following:
- **Write** – Sets permission to write, delete, or create all of the application's files.
 - **Read** – Sets permission to read all of the application's files.
 - **Find** – Sets permission to search for any of the application's files.
 - **Execute** – Sets permission to execute any of the application's files.

NOTE

The deny settings will always take precedence over the allow settings, which means that setting both will result only in the deny action.

- 6) Click **Apply** to set the new security override settings for the specified process. Multiple processes can be added to the security override list by repeating the steps. Each process can also be modified or removed as needed.

Chapter 7: Modifying Project Assets

At various times it may be necessary to make modifications to the project, including adding, deleting, or changing appset assets, or removing *artifacts* (unwanted DLLs, etc.) that were introduced as a side effect of capturing the application installation. This can be easily done using the various view tabs in **Cloudpaging Studio: Files, Registry, Environment, Fonts, Startup, and Services & Drivers**.

NOTE

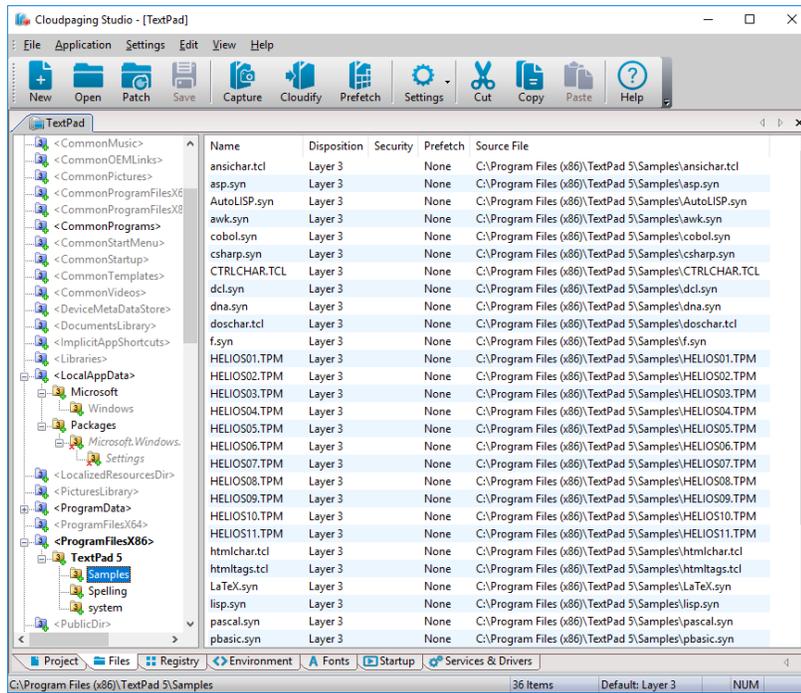
The material in this chapter applies to “cleaning” up the appset (Step 5) in [Chapter 3, Capturing the Application Installation](#), preparing an appset for cloudifying in [Chapter 4, Cloudifying the Application](#) and to modifying a patched appset in [Chapter 9, Patching an Appset](#).

The following are explained in this chapter:

- Modifying Files and Folders
- Modifying Registry Entries
- Modifying Environment Variables
- Modifying Fonts
- Modifying Startup Items
- Modifying Services and Drivers
- Modifying Shortcuts
- Searching for Assets
- Globally Changing Asset Dispositions
- Understanding Color Coding in Views

Modifying Files and Folders

The Files view is accessed by selecting **Files** from the **View** menu, or by selecting the **Files** view as shown below.



Most commands can be accessed either from the **Edit** menu or by right-clicking a folder in the folders (left) pane, or by right-clicking a file in the files (right) pane.

NOTE

Security descriptors are single letters based on the keyboard shortcuts in the File property dialog.

For example, R = Make Read-Only.

About Folders

There are the two types of folders that appear in the folders (left) pane of the Files view. These folders are colored to make identification simple. All the parent folders in the view are considered “root” folders to the project. These root folders correspond to various locations on the real file system.

Root and Template Folders

- **Root folders** – The yellow-colored folders are folders that were captured during installation or manually added to the top of the folder pane (left) of the Files tree view. To avoid folder conflicts on the Target PC, you should install the applications being captured to a Root folder, and to give this folder a unique name.
- **Template folders** – All the blue-colored folders have unique names called *templates* that correspond to *special folders* in Windows. *Templates* are aliases that map to actual system folder paths (e.g., *WindowsDir* typically maps to “c:\windows”). The specific system folder that a template maps to depends on the version of the Windows operating system, the particular operating system installation (e.g., if Windows is installed on the D drive, then *WindowsDir* will map to “d:\windows”),

the logged-in user, or a combination thereof. So, templates allow for the Target PC to have root folders in different locations, making it possible for an appset to “adapt” to the Target PC (e.g., if a user moves the “My Documents” folder to a different location, the corresponding template will map to that new location when the appset is initially paged). Template folders cannot be removed or added to the view. Empty templates folder (shown with gray text) will not be added to the final cloudified application.

NOTE

More information on folder mappings (called “special folders” under Windows) is available online at <https://support.microsoft.com/en-us/kb/194702>

Important

The “Documents and Settings” folder is not a system root folder since Windows does not support a template mapping for it. Because of this, be sure this folder is **NOT** included in the appset as an Installed Folder. The sub-folders in “Documents and Settings” all have templates.

Merged and Non-merged Folders

Merged folders are identified by a green plus sign (“+”) over the folder icon. These are typically folders that exist on a real system. Files or sub-folders within these folders will be *merged* with the contents on the Target PC during the virtualization phase. If there is a name conflict with a real sub-folder or file on the Target PC, only the paged version will be visible. These folders will also be merged with corresponding folders in other paged applications.

Non-merged folders set to virtual-integrated (layer-3) are still merged from the Target PC’s system view. However, from the virtual application’s point-of-view, the folder and content are not merged. The virtual application will only see the contents of these unmerged folders, ignoring the contents of the folder on the Target PC and merged folders from any other appsets. Non-merged folders set to virtual isolated (layer-4) are invisible to everything except the virtual application.

The examples below will help clarify these concepts.

Merged Folder Visibility

This section illustrates how an appset’s merged folders get merged with the real folders of the Target PC and the folders of other appsets. The system will always have a merged view on content that it has access to (e.g., layer 3 content). Merged rules only apply to the virtual applications.

Suppose that appsets A, B, C and D have been paged on the Target PC, in that order, and that the system has a real folder X;

- appset A has a virtual-integrated (layer 3) merged folder X;
- appset B has a virtual-integrated (layer 3) merged folder X;
- appset C has a virtual-isolated (layer 4) merged folder X;
- appset D has a virtual-isolated (layer 4) non-merged folder X;
- appset E has no folder X.

For the purpose of this discussion, we only consider the case where the folder X contents in appsets are virtual-integrated (layer 3) or virtual-isolated (layer 4) since layer 1 and layer 2 content is treated just like system content. For more information about disposition layers see [Editing Folder Properties](#).

The table below shows the visibility of folder X's content to the system and the various appsets. Each list in the left column shows the visibility priority in case of conflicting content. For instance, the first row indicates that

- the system will see non-conflicting content from B, A, and itself;
- conflicting content from B is visible to the system and hides all other;
- content from A that conflicts with the system, but not with B is visible to the system and hides the system's own content.

This...	Sees X Content from...
System	B, A, System
Appset A	A, B, System
Appset B	B, A, System
Appset C	C, B, A, System
Appset D	D
Appset E	B, A, System

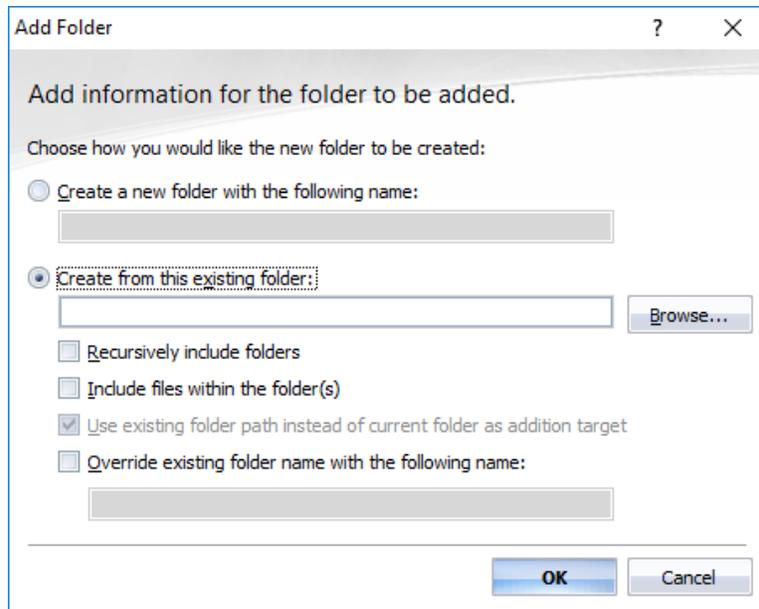
- conflicting content between A and B results in A viewing A's own content and B viewing B's content, while the system views B's content.

For information about making a folder merged see [Making Root Folders Merged](#).

Adding Root Folders

To add a root folder:

- 1) Right-click any folder or on a blank area of the folders pane (left).
- 2) Select **Add Root Folder** from the pop-up menu. The **Add Root Folder** dialog appears.



- 3) If you wish to create a new root folder, select **Create new root folder with the following name** and enter a path for it. If you wish to add an existing folder instead, select **Create from existing source folder**.
- 4) If you chose **Create from existing source folder**, you can also select one or more of the following options:
 - **Recursively include subdirectories** – Selecting this option will cause all subfolders under the selected folder to be added as well. Otherwise, only the selected folder is added.
 - **Include files within the directory(s)** – Selecting this option will add all the files in the folders being added. Otherwise, only the folders are added.
 - **Use existing folder path instead of current folder as addition target** – This option is automatically selected because you are adding a root folder and not a subfolder.
 - **Override existing folder name with the following name** – Normally, the selected folder is added with its current name. If you wish to add the folder under a different name, clear this option and specify the new name to be given to the folder.
- 5) Click **OK**. The Files view will be updated to reflect the added folders and files.

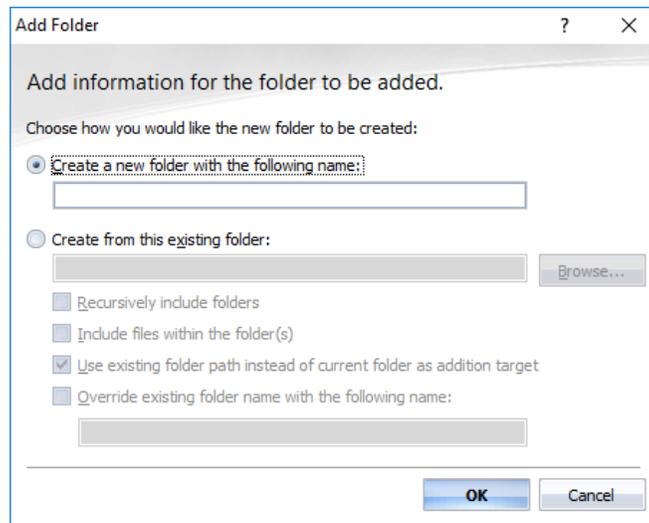
Adding Sub-folders

To add a sub-folder

- 1) To add a sub-folder, right-click any root folder or folder template.
 - A)
 - B)



2) Select **Add SubFolder** from the pop-up menu. The **Add Folder** dialog displays.



3) If you wish to create a new folder, select **Create new folder with the following name** and enter a name for it. If you wish to add an existing folder instead, select **Create from this existing folder**.

4) If you chose to add an existing folder, you can also select one or more of the following options:

- **Recursively include subdirectories** – Selecting this option will cause all subfolders under the selected folder to be added as well. Otherwise, only the selected folder is added.
- **Include files within the directory(s)** – Selecting this option will add all the files in the folders being added. Otherwise, only the folders are added.
- **Use existing folder path instead of current folder as addition target** – Instead of adding the new folder as a subfolder of the selected folder, the complete path of the new folder will be added to the project. This saves you from having to manual recreate the entire path, starting with the root folder.

- **Override existing folder name with the following name** – Normally, the selected folder is added with its current name. If you wish to add the folder under a different name, clear this option and specify the new name to be given to the folder.
- 5) Click **OK**. The Files view will be updated to reflect the added subfolders and files.

Renaming Folders

Any non-template folder can be renamed.

To rename a folder:

- 1) Right-click the folder that you wish to rename.
- 2) Select **Rename** from the pop-up menu.
- 3) Rename the folder and press **Enter** once you are finished.

Editing Folder Properties

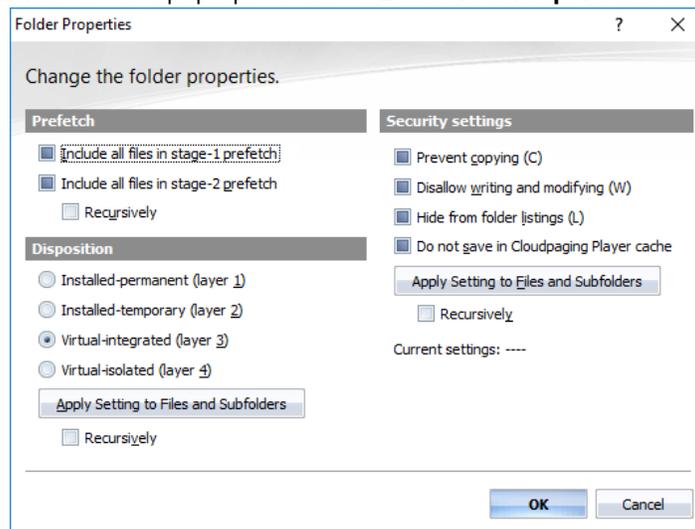
A file has a number of security and prefetch settings that can be modified to suit your specific needs.

To edit folder properties:

Important

For some of the settings discussed below, if **Recursively** is selected, all sub-folders and files within them will be affected.

- 1) Right-click the folder with properties you wish to change.
- 2) Select **Folder Properties** from the pop-up menu. The **Edit Folder Properties** dialog appears.



- 3) Configure the following settings as required:
 - Prefetch - Select Include all files in Stage-1 prefetch or Include all files in Stage-2 prefetch. For more information, see [Prefetching](#).
 - Disposition - The layer number displayed on the folder indicates the disposition setting. Select one of these disposition settings:

- **Installed-permanent (Layer 1)** - Copies assets (files and keys) permanently onto the local system, and can be seen by the entire local system.
- **Installed-temporary (Layer 2)** - Installs assets during the activation process, and uninstalls assets during the deactivation process. The original asset is backed up before the new asset is installed, and when the new is uninstalled, the original asset is restored.
- **Virtual-integrated (Layer 3)** - (default setting) Assets that can be seen by both the paged application and the local system, but are not physically installed on the local system.
- **Virtual-isolated (Layer 4)** - Assets that can only be seen by the paged application and are not physically installed on the local system.

Important

Layer 1 or layer 2 files cannot have security settings applied to them. In addition, **Cloudpaging Player** will not track usage against files that are layer 1 or layer 2.

NOTE

Layer 1 and layer 2 options are disabled for non-merged folders because folders with layer 1 or layer 2 dispositions are always merged. See [Merged and Non-merged Folders](#).

- **Security Settings** - Select one of these settlings:

Security Setting	Description	File enabled by default
Deny Reading and Copying	Assets cannot be copied	Command-line file
Deny Writing and Modifying	Assets can not be altered	Command-line file
Hide from folder listings	Assets are not displayed in listing of folders	(none)
Do not cache on local machine	Assets are not permitted be cached to the local system	(none)

- 4) Click **OK**. The Files view will be updated to reflect the changes made.

Setting the Working Folder

The **Working Folder** is the folder that contains the application's executable or some related files. A vast majority of the time, using the same directory as used in the command line will work. On occasion using the same directory as where the program executable is found will not work. In these cases, it may be required to set the Working Directory to the application root directory. The easiest way to populate this field is to navigate in the Files view to the folder that you want to set as the working folder, right-click it, then select **Set As Working Folder**.

Making Root Folders Merged

By default, root folders are virtual and their contents will not be merged with folders already on the system.

See [Merged and Non-merged Folders](#)

To merge a Root folder with system folder:

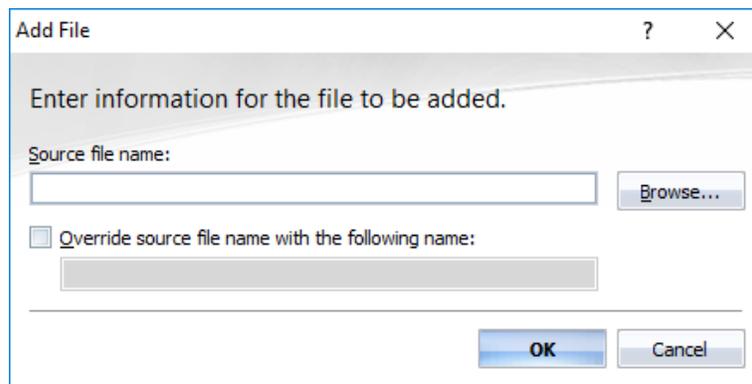
- 1) Right-click the Root folder you wish to change.
- 2) Check the **Merged Folder** option.
- 3) A green plus sign (“+”) will appear over the folder symbol.

Adding Files

Cloudpaging Studio allows you to add any files to any folder.

To add a file to a folder:

- 1) Right-click the desired folder or click a blank area of the Files view if the desired folder is currently selected.
- 2) Select **Add File** from the pop-up menu. The **Add File** dialog appears.



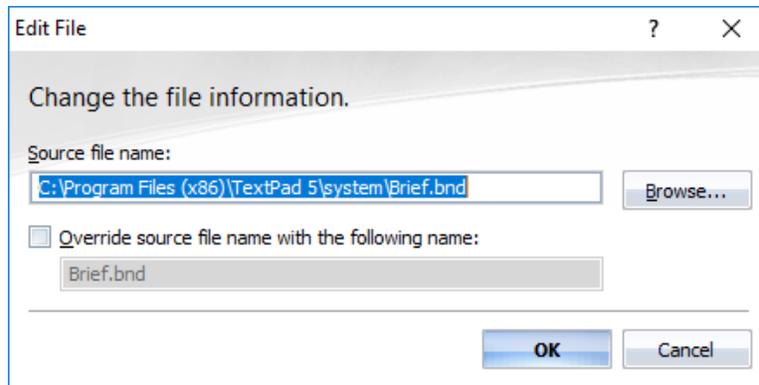
- 3) Enter the path of the file you wish to add or browse to it.
- 4) Normally, the selected file is added with its current name. If you wish to add the file under a different name, select **override source file name with the following name**.
- 5) Enter the new file name.
- 6) Click **OK**. The Files view will be updated to reflect the added file.

Editing Files

If after adding a file you realize that you have added the wrong file or you decide that you wish to change its name, you can do so without the need to delete the file and add it again.

To change the source or name of a file:

- 1) Right-click the file whose source or name you wish to change.
- 2) Select **Edit File** from the pop-up menu. The **Edit File** dialog appears.



- 3) To change the source of the file, enter the new path of the source file or browse to it.
- 4) To change the name of the file, make sure the **Override source file name with the following name** option is not selected, and specify the new file name.
- 5) Click **OK**. The Files view will be updated to reflect the changes that were made.

Editing File Properties

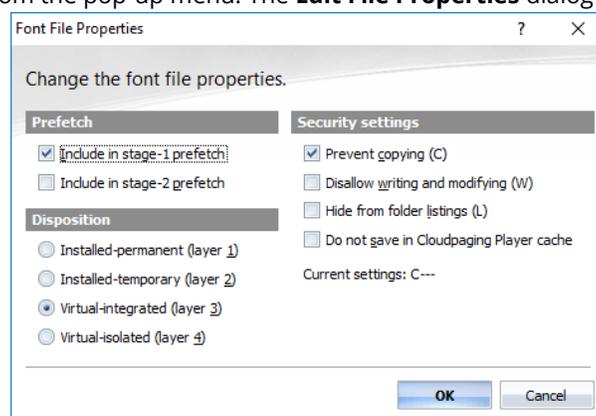
A file has a number of security and prefetch settings that can be modified to suit your specific needs.

To edit file properties:

Important

For some of the settings discussed below, if **Recursively** is selected, all sub-folders and files within them will be affected.

- 1) Right-click the file that has properties you wish to change.
- 2) Select **File Properties** from the pop-up menu. The **Edit File Properties** dialog appears.



- 3) Configure the following settings as required:
 - Prefetch - Select Include in stage-1 prefetch or Include in stage-2 prefetch. For more information see [Prefetching](#).
 - Disposition - See [Editing Folder Properties](#).
 - Security Settings - See [Editing Folder Properties](#).

Important

Security settings are **not** enabled by default. Please add them to files you wish to protect.

Deleting Files and Folders

It is possible to delete any file or root folder that was created during installation or that was added manually.

To delete a file or folder:

- 1) Right-click a file or folder that you wish to delete.
- 2) Select **Delete** from the pop-up menu.
- 3) Select **OK** when prompted for confirmation.
- 4) If you wish to delete another file or folder, repeat steps 1-3.

Excluding Files and Folders

Excluded folders, sub-folders, and files are not included when the appset is cloudified but are still part of the project. This gives you the flexibility of including certain folders and files for one deployment of an appset and not for another. Excluding folders and files also enables you to test an appset without a folder or file before permanently deleting it from the project.

To exclude a file or folder:

- 1) Right-click a file or folder that you wish to exclude.
- 2) Select **exclude** from the pop-up menu.
- 3) Excluded folders will have a red X over their symbol. Excluded files are shown with their name and properties in italicized text.
- 4) If you wish to exclude another file or folder, repeat steps 1-3.

Setting the Command Line

The **Command Line** is the path and executable file that are used to start the program.

To set the Command Line:

- 1) In the Files view, navigate to the application's executable, and right-click it.
- 2) Select **Set as command line**.
 - If you are not sure where the executable is located, one way it can be found is by looking at the properties of the shortcut created when the application was installed.
 - If you want to launch a **non-.exe** file such as a **.doc** file, it is possible. To cloudify non-applications, the command line must be set to specify an executable to be launched (such as an .exe or .com file). Then the name of the data file can be specified. For example: `"%SystemRoot%\System32\mspaint.exe" "C:\My Images\Image.bmp"`.
 - If the file for the Command Line is a shortcut (*.lnk file), then the target of the shortcut will be used.

Locating Files/Folders in Windows Explorer

If for any reason you need to examine a file or folder in its physical location on the Cloudifying PC, **Cloudpaging Studio** can display the file or folder in Windows Explorer directly from the Files view.

To show a file in Explorer:

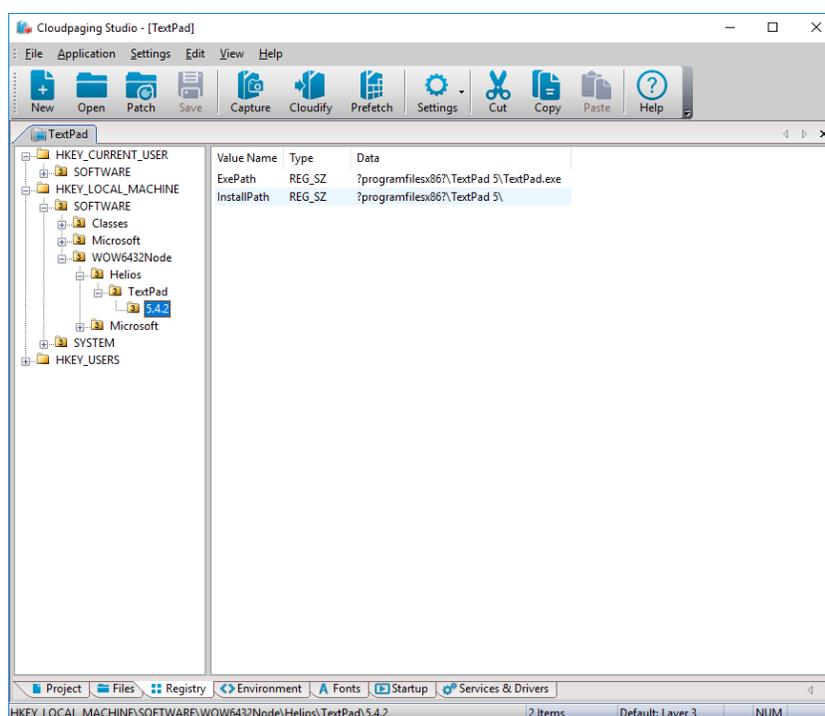
- 1) Right-click on the file asset in the right-hand pane of the Files view.
- 2) Select Show Source File in Explorer.
- 3) Windows Explorer will open to the correct folder with the selected file highlighted.

To show a folder in Explorer:

- 1) Right-click on the folder in the left-hand pane of the Files view.
- 2) Select **Show in Explorer**.
- 3) Windows Explorer will open to the correct folder.

Modifying Registry Entries

The Registry view is accessed by selecting **Registry** from the **View** menu, or by selecting the **Registry** view as shown below.



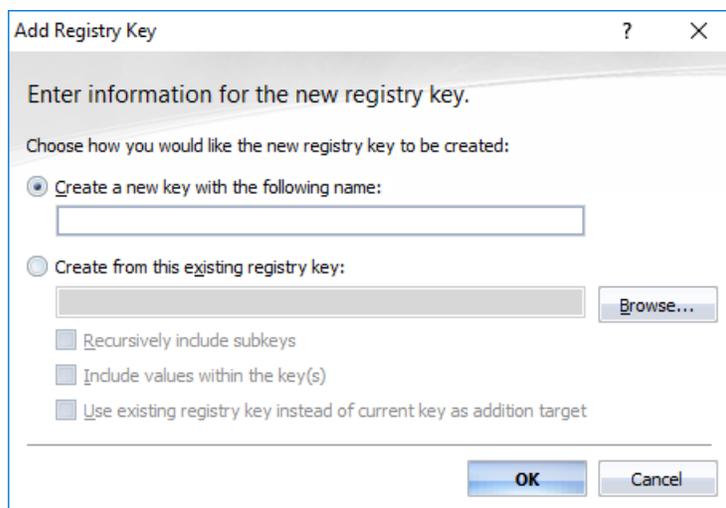
Most commands can be accessed either from the **Edit** menu or by right-clicking a key in the registry keys pane (left), or right-clicking a value in the registry values pane (right).

Adding Registry Keys

You can create a new subkey or add an existing subkey under any key in the Registry view, including *root keys* (or *hives*). However, you cannot create new root keys.

To add a subkey

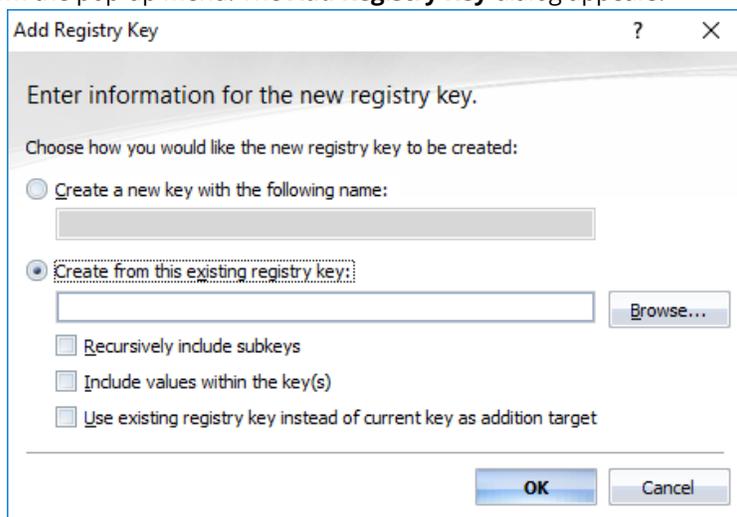
- 1) Right-click any key in the registry keys pane (left).
- 2) Select **Add Key** from the pop-up menu. The **Add Registry Key** dialog appears.



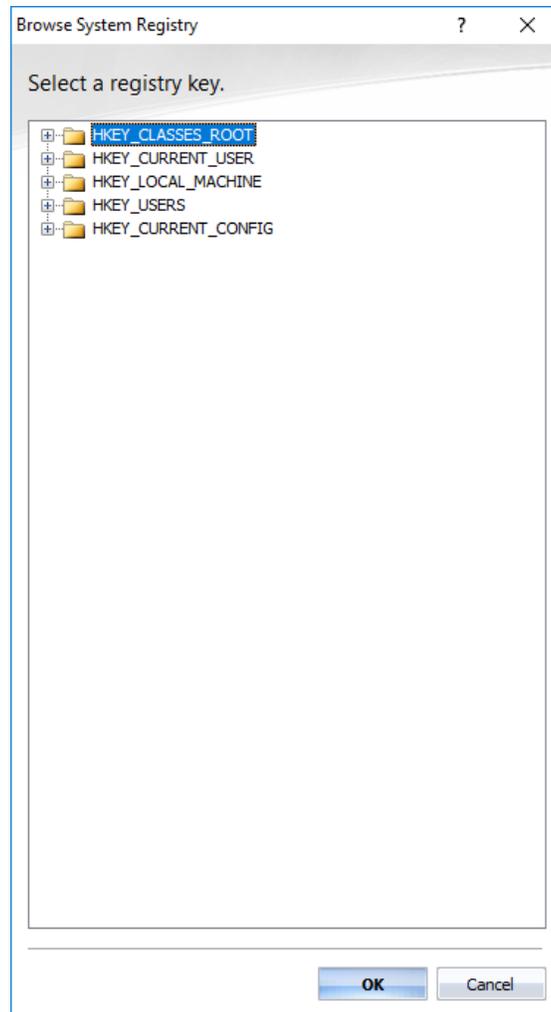
- 3) Enter a name for the new subkey. If you wish to add an existing key instead, select **Create from this existing registry key**.
- 4) Click **OK**.
 - The Registry view will be updated to reflect the added subkeys and values.

To add an existing subkey

- 1) Right-click any key in the registry keys pane (left).
- 2) Select **Add Key** from the pop-up menu. The **Add Registry Key** dialog appears.



- 3) Select Create from this existing registry key.
- 4) Click **Browse...**
- 5) The Browse System Registry dialog appears.



- 6) Locate and select the desired key.
- 7) Click **OK**.
- 8) You can then select one or more of the following options:
 - **Recursively include subkeys** – Selecting this option will cause all subkeys under the selected key to be added as well. Otherwise, only the selected key is added.
 - **Include values within the key(s)** – Selecting this option will add all the values in the keys being added. Otherwise, only the keys are added.
 - **Use existing registry key instead of current key as addition target** – Instead of adding the new key as a subkey of the selected key, the complete path of the new key will be added to the project, starting from top level. This saves you from having to manually recreate the entire path, starting with the top level.
- 9) Click **OK**.

Renaming Registry Keys

To rename a registry key

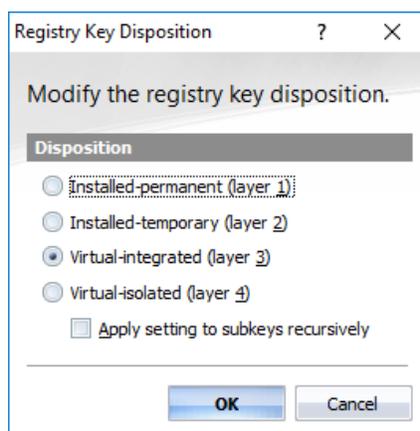
- 1) Right-click the registry key that you wish to rename.
- 2) Select **Rename** from the pop-up menu.

- 3) Rename the folder and press **Enter** once you are finished.

Changing the Disposition of a Registry Key

To change the disposition of a registry key:

- 1) Right-click the desired key, and select **Disposition** from the pop-up menu. The **Registry Disposition** dialog appears.



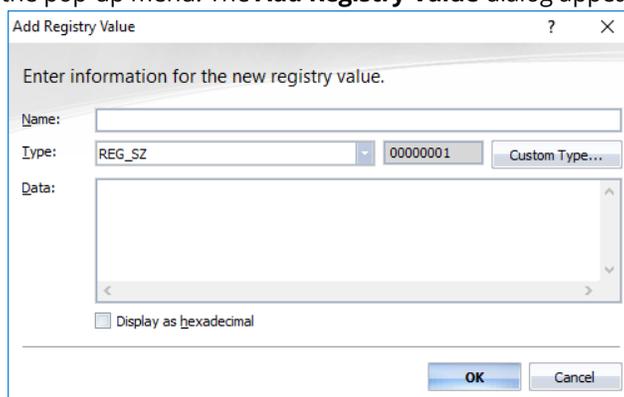
- 2) Select the disposition. For more information about disposition settings, see [Editing Folder Properties](#).
- 3) If you want that disposition to apply to all subkeys, click **Apply setting to subkeys recursively**.
- 4) Click **OK**.
 - The Registry view will be updated to reflect the changed disposition setting for the selected key.

Adding Registry Values

Cloudpaging Studio allows you to add values to any key.

To add a registry value to a registry key:

- 1) Right-click the desired key or on a blank area of the Registry values pane if the desired key is currently selected.
- 2) Select **Add Value** from the pop-up menu. The **Add Registry Value** dialog appears.



- 3) Enter the name of the value you wish to add in the **Name** field.
- 4) Select the data type of value from the **Type** drop-down list.
- 5) If you wish to view or enter the data in hexadecimal format, check the **Display as hexadecimal** option. Otherwise, data is assumed to be in decimal format.

- 6) Enter the data in the **Data** field.
- 7) Click **OK**. The Registry view will be updated to reflect the added value.

NOTE

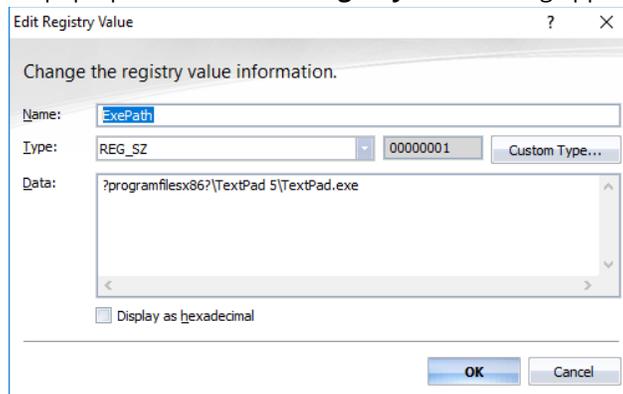
If the registry value appears to contain an environment variable, such as %ServerName%, Cloudpaging Studio will attempt to resolve the environment variable on the Cloudifying PC. If there is no such environment variable, Cloudpaging Studio will conclude that the variable name is a constant and will escape the “%” characters. %ServerName% will become %%ServerName%. Cloudpaging Studio will log these changes in the Cloudpaging Studio log file located in the installation directory. In order to preserve the environment variable in the cloudified application, define the environment variable on the Cloudifying PC before creating the appset.

Editing Registry Values

You can edit registry values after they have been added.

To edit a registry value:

- 1) Right-click the value you wish to edit.
- 2) Select **Edit Value** from the pop-up menu. The **Edit Registry Value** dialog appears.



- 3) Make the necessary changes. For more information, see [Adding Registry Values](#).
- 4) Click **OK**. The Registry view will be updated to reflect your changes.

Deleting Registry Keys and Values

It is possible to delete any registry key or value that was created during installation or that was added manually.

To delete a registry key or value:

- 1) Right-click the registry key or value that you wish to delete.
- 2) Select **Delete** from the pop-up menu.
- 3) Select **OK** when prompted for confirmation.

Excluding Registry Keys

Excluded registry keys are not included when the appset is cloudified but are still part of the project. This gives you the flexibility of including certain registry keys for one deployment of an appset and not for another.

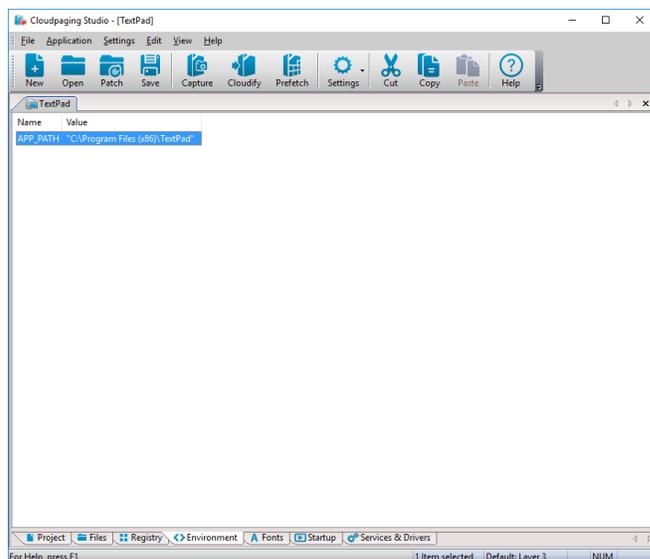
Excluding registry keys also enables you to test an appset without a registry key before permanently deleting it from the project.

To exclude a Registry Key:

- 1) Right-click a registry key that you wish to exclude.
- 2) Select **Exclude** from the pop-up menu.
- 3) Excluded registry keys are shown with their name and properties in italicized text.
- 4) If you wish to exclude another registry key, repeat steps 1-3.

Modifying Environment Variables

The Environment view is accessed by selecting **Environment** from the **View** menu, or by selecting the **Environment** view as shown below.



This view displays environment variables that have been created either during installation or manually added. Most relevant commands can be accessed either from the **Edit** menu or by right-clicking on an environment variable.

There are two types of environment variables: user environment variables (specific for each user and set in the registry under HKEY_CURRENT_USER \ Environment) and system environment variables (global for everyone and set in the registry under HKEY_LOCAL_MACHINE \ SYSTEM \ CurrentControlSet \ Control \ Session Manager \ Environment). When an application installation is captured, **Cloudpacing Studio** will record the changes to the environment variables from both the user and system locations and merged the results into the Environment view. These environment variables are then injected into either the application or to the system depending on the **Environment Variables** setting under the **Virtualization** screen on the **Settings** dialog. See [Virtualization Settings](#).

- Environment variables shared with the entire system will be added to the HKEY_CURRENT_USER \ Environment registry key when the appset is virtualized making them available to all applications for that specific user.
- Environment variables visible only to the application are set in the process environment for the application when it is launch from **Cloudpacing Player**. Child-processes of the application may inherit these environment variables but no other process on the system can access them.

All virtual environment variables will override any existing variables of the same name for either method. The only exception to this rule is for the %PATH% or %PATHEXT% variables. These will automatically be appended to exists system values much like:

PATH = %PATH%;c:\newfolder

Visit <http://support.microsoft.com/kb/104011> for more general information on setting environment variables.

NOTE

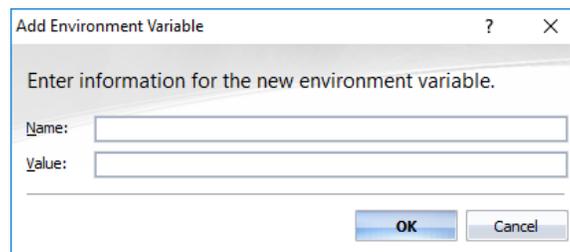
Some legacy applications may check the registry for environment variables under a specific hive and not use the variables in the process environment. For such applications, it may be necessary to add the environment variables directly into the Registry view in the correct locations and set the key disposition to Layer 4.

Adding Environment Variables

You can create new environment variables as described below.

To add an environment variable:

- 1) Right-click a blank area of the Environment view.
- 2) Select **Add Environment Variable** from the pop-up menu.



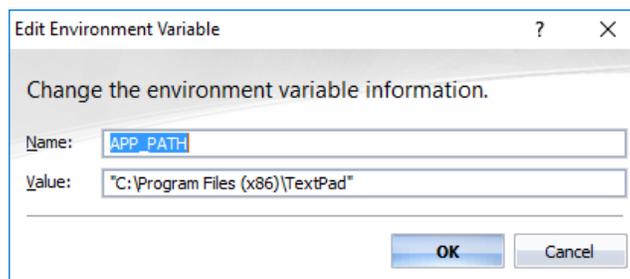
The **Add Environment Variable** dialog appears.

- 3) Enter a name for the new environment variable in the **Name** field.
- 4) Enter a value for the variable in the **Value** field.
- 5) Click **OK**. The Environment view will be updated to reflect the added variable.

Editing Environment Variables

To edit a System environment variable:

- 1) From the Environment view, right-click the environment variable that you wish to edit.
- 2) Select **Edit Environment Variable** from the pop-up menu.



The **Edit Environment Variable** dialog appears.

- 3) Make the necessary changes. For more information, see [Adding Environment Variables](#).

- 4) Click **OK**. The Environment view will be updated to reflect your changes.

Deleting Environment Variables

It is possible to delete any environment variable that was created during installation or that was added manually.

To delete an environment variable:

- 1) Right-click the environment variable that you wish to delete.
- 2) Select **Delete** from the pop-up menu.
- 3) Select **OK** when prompted for confirmation.

Excluding Environment Variables

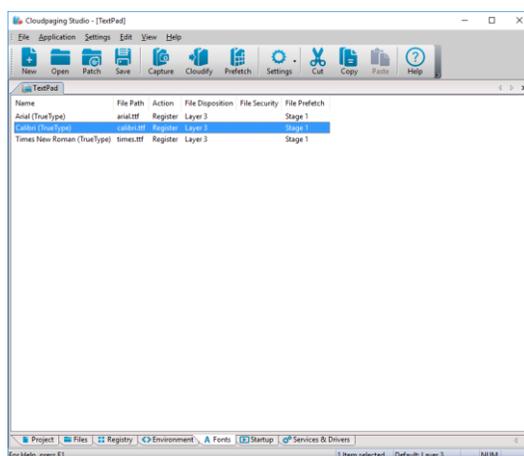
Excluded environment are not included when the appset is cloudified but are still part of the project. This gives you the flexibility of including certain environment variables for one deployment of an appset and not for another. Excluding environment variables also enables you to test an appset without an environment variable before permanently deleting it from the project.

To exclude a Environment Variable:

- 1) Right-click an environment variable that you wish to exclude.
- 2) Select **Exclude** from the pop-up menu.
- 3) Excluded environment variables are shown with their name and properties in italicized text.
- 4) If you wish to exclude another environment variable, repeat steps 1-3.

Modifying Fonts

Fonts installed during the capture process will be shown on the Fonts view. You can manually add other fonts to a project. Most relevant commands can be accessed either from the **Edit** menu or by right-clicking on a font asset.



The fonts in the appset will either be available only to the appset (injected) or made available to other applications on the Target PC (registered), according to the Font setting under the Virtualization screen on the **Settings** dialog. See [Virtualization Settings](#).

NOTE

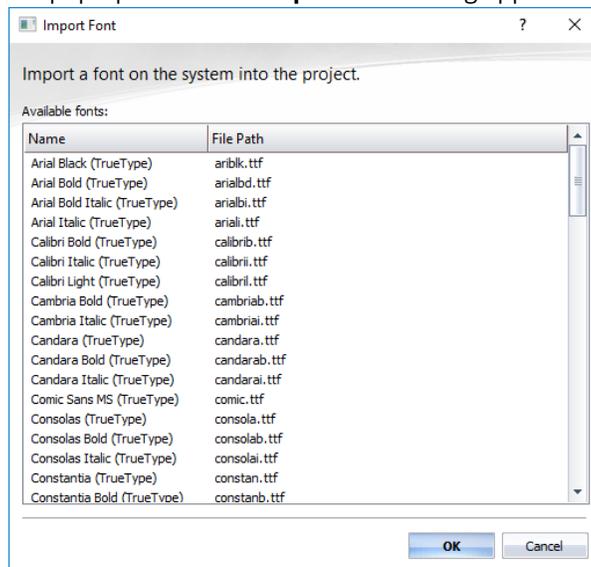
To add more fonts to a project, you will need to know the file name or names for the font, and not just the font name. Keep in mind that in many cases, there is more than one file for each font. For example, "Arial" consists of four files, one for the basic font, another for italic, bold, and bold italic.

Adding Fonts

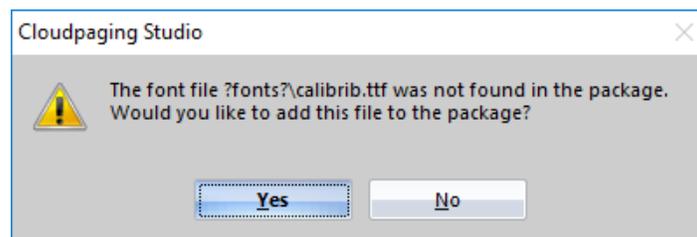
There are two ways to add a font to the project: Importing system fonts or selecting the font files directly. This second option is useful if adding fonts that are not installed on the Cloudifying PC.

To import a system font into a project:

- 1) Right-click a blank area of the Fonts view.
- 2) Select **Import Font** from the pop-up menu. The **Import Font** dialog appears.



- 3) Select the desired system font.
- 4) Click **OK**.



If the corresponding font file is found on the Cloudifying PC, **Cloudpacing Studio** will prompt you, asking if you want to add the file to the project.

This may not be necessary if you are confident the font will be present on the Target PC.

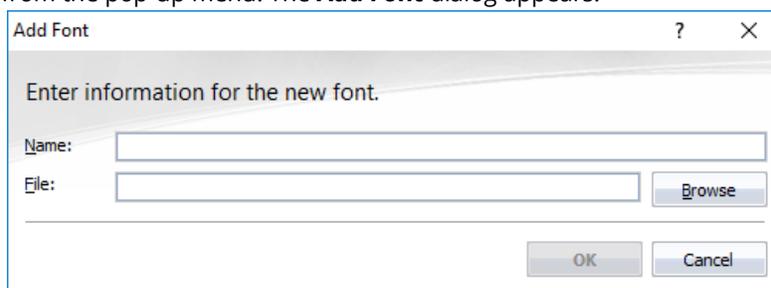
If the selected file is not found on the Cloudifying PC, then **Cloudpacing Studio** will alert you, asking if you still wish to add the font to project.

If the font file is not added to the project, the file path will be shown in red in the Fonts view.

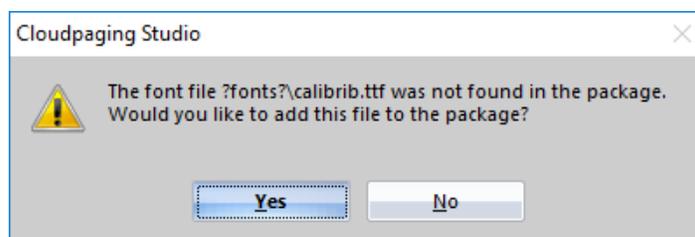
- 5) The Fonts view will be updated to reflect the added font.

To manually add a font to a project:

- 1) Right-click a blank area of the Fonts view.
- 2) Select **Add Font** from the pop-up menu. The **Add Font** dialog appears.



- 3) Enter a name for the new font in the **Name** field.
- 4) Click **Browse** and locate the font file in the **File** field.
- 5) Click **OK**.



If the corresponding font file is found on the Cloudifying PC, **Cloudpaging Studio** will prompt you, asking if you want to add the file to the project. This may not be necessary if you are confident the font will be present on the Target PC. If the selected file is not found on the Cloudifying PC, then **Cloudpaging Studio** will alert you, asking if you still wish to add the font to project. If the font file is not added to the project, the file path will be shown in red in the Fonts view.

- 6) The Fonts view will be updated to reflect the added font.

NOTE

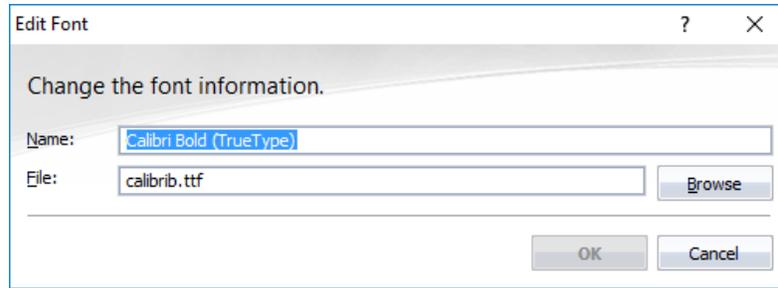
If the virtualization setting for fonts is set to register, but the disposition for the corresponding font file is set to virtual-isolated (layer 4), the disposition for the font will be shown in red. This font will not be visible to other applications on the Target PC, despite the virtualization setting. For more information about conflict alerts, see [Conflict Color Coding](#).

Editing Fonts

Existing fonts can be edited, their name or the file changed if necessary.

To edit a font:

- 1) From the Fonts view, right-click the font that you wish to edit.
- 2) Select **Edit Font** from the pop-up menu. The **Edit Font** dialog appears.



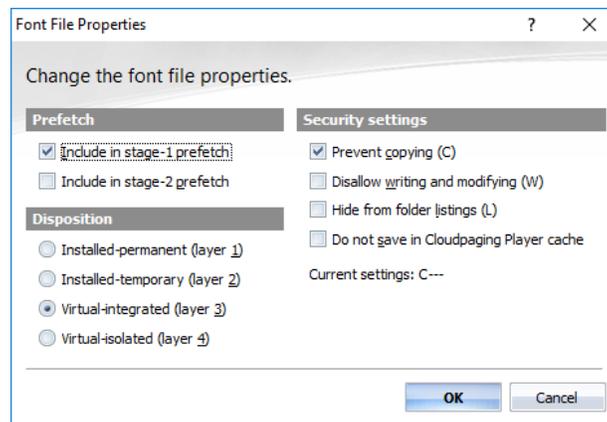
- 3) Make the necessary changes. For more information, see [Adding Fonts](#).
- 4) Click **OK**. The Fonts view will be updated to reflect your changes.

Editing the File Properties of a Font

Adding a Font asset also adds the file to the Files view. You can edit the prefetch, disposition, and security settings from within the Fonts view, rather than having to locate and edit the file from the Files view.

To edit the file properties of a Font asset:

- 1) Right-click the asset with properties you wish to edit.
- 2) Select **Edit File Properties** from the pop-up menu. The **Font File Properties** dialog appears.



- 3) Make the necessary changes. For more information, see [Editing File Properties](#).
- 4) Click **OK**. The Fonts view will be updated to reflect the changes.

Deleting Fonts

It is possible to delete any font that was created during installation or that was added manually.

To delete a font:

- 1) Right-click the font that you wish to delete.
- 2) Select **Delete** from the pop-up menu.
- 3) Select **OK** when prompted for confirmation.

Excluding Fonts

Excluded fonts are not included when the appset is cloudified, but are still part of the project. This gives you the flexibility of including certain fonts for one deployment of an appset and not for another. Excluding fonts also enables you to test an appset without a font before permanently deleting it from the project.

To exclude a Font:

- 1) Right-click a font that you wish to exclude.
- 2) Select **Exclude** from the pop-up menu.
- 3) Excluded fonts are shown with their name and properties in italicized text.
- 4) If you wish to exclude another font, repeat steps 1-3.

Locating the File or Registry Entry of a Font

If you need to view the Files View entry or the registry key for the selected font, **Cloudpaging Studio** will take you to the correct asset directly from the Fonts View.

Locating the font file in the Files View

- 1) Right-click the font for which you wish to view the command file.
- 2) Click **Jump To** and select **Font File**.
- 3) **Cloudpaging Studio** will switch to the Files View, select the correct folder and highlight the appropriate file.

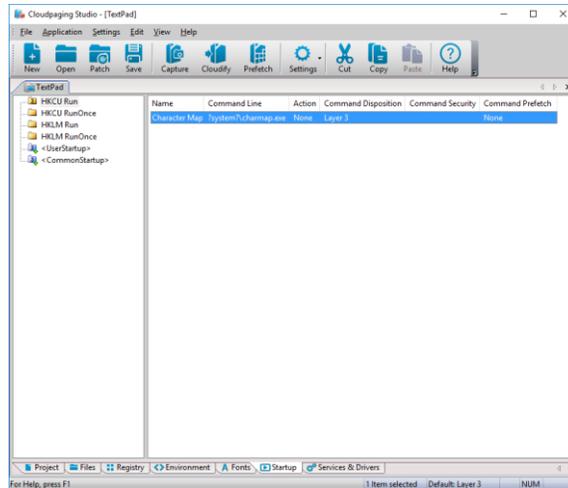
Locating the registry value in the Registry View

- 1) Right-click the font for which you wish to view the registry value.
- 2) Click **Jump To** and select **Fonts Value**.
- 3) **Cloudpaging Studio** will switch to the Registry View, select the correct key, and highlight the value.

Modifying Startup Items

Startup items are programs that typically launch at system startup, which are installed along with the paged application. These often include the system tray icons that permit direct access to the functions of the paged application, even when the paged application is not running.

Any startup items created during installation will be visible in the Startup view. Others can be added, and any item can be deleted or excluded.



Types of Startup Items

Windows launches Startup items using one of two different methods: Registry “Run” Key items and Start Menu items. Both kinds of captured startup items can be either enabled or disabled via the **Virtualization** screen on the **Settings** dialog. See [Virtualization Settings](#).

Registry Run Key Items

The path to the executable is found in the value of special registry keys called *Run Keys*. These startup items are found in one of the yellow folders on the **Startup** view:

- **HKCU Run** (“HKCU” is short for “HKEY_CURRENT_USER”) – These startup items apply only to the currently logged-in user.
- **HKCU RunOnce** – Similar to HKCU Run items, but these startup items will only run once. If one of these items has already been loaded before the cloudified application is virtualized on the Target PC, it will not load again.
- **HKLM Run** (“HKLM” is short for “HKEY_LOCAL_MACHINE”) – These startup items apply to all local users on the Target PC.
- **HKLM RunOnce** – Similar to the “HKCU RunOnce” startup items in that they will only load once and not every time the cloudified application is virtualized on the Target PC. Like the “HKLM Run” startup items, these apply to all local users on the Target PC.

NOTE

Adding a startup item to the registry does not add the corresponding executable file to the project. This may not be necessary if you are confident that the executable will be present on the Target PC. If the corresponding file is not present in the File view, the path will be shown in red in the Startup View. For more information about conflict alerts, see [Conflict Color Coding](#).

Start Menu Items

These items appear as shortcuts in the **Startup** submenu under the Programs Start Menu. These items are found in the two blue folders on the **Startup** view:

- **UserStartup** – These Startup items apply only to the currently logged-in user.
- **CommonStartup** – These Startup apply to all local users on the Target PC.

NOTE

Adding a startup item to the start menu does not add the corresponding executable file to the project. This may not be necessary if you are confident that the executable will be present on the Target PC. If the corresponding file is not present in the File view, the path will be shown in red in the Startup View. For more information about conflict alerts, see [Conflict Color Coding](#).

Adding Startup Items

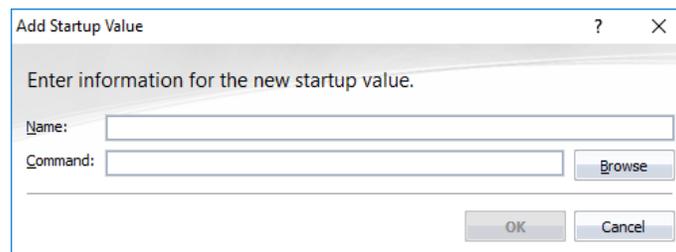
Before adding a Startup Item, you need to have the following information:

- Registry or Start Menu
- Name and location of the registry key (if necessary).
- Location and name of the executable file.

There are two different methods, depending on the type of Startup item being added.

To add a registry key Startup item:

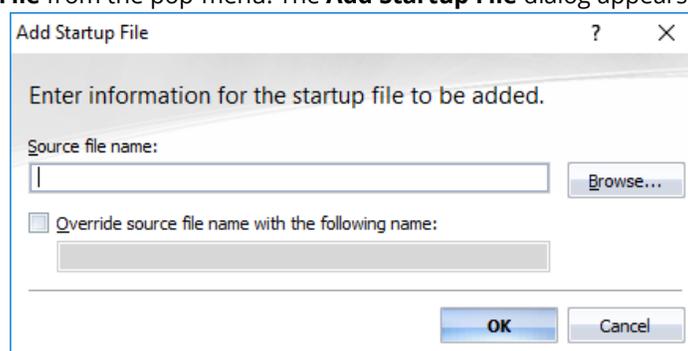
- 1) Right-click either the HKCU or the HKLM folder, or select one of these folders and right-click in a blank area within the item view.
- 2) Select **Add Startup Item** from the pop-menu. The **Add Startup Value** dialog appears.



- 3) Enter the **Name** of the new Registry Key.
- 4) In **Command**, enter the full path to the executable file.
- 5) Click **OK**.

Adding Start Menu Startup Items

- 1) Right-click either the UserStartup or the CommonStart folder, or select one of these folders and right-click in a blank area within the item view.
- 2) Select **Add Startup File** from the pop-menu. The **Add Startup File** dialog appears.



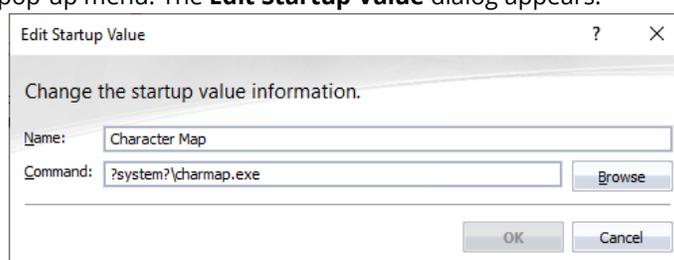
- 3) Enter the path of the file you wish to add or browse to it.
- 4) Normally, the selected file is added with its current name. If you wish to add the file under a different name, select **override source file name with the following name**.
- 5) Enter the new file name.
- 6) Click **OK**. The Startup view will be updated to reflect the added file.

Editing Startup Items

Editing Startup items can also be done from the File or Registry views, depending on the type, but it is usually simpler to edit them directly from the Startup view.

Editing Registry Key Startup Items

- 1) Right-click the item with properties you wish to edit.
- 2) Select **Edit** from the pop-up menu. The **Edit Startup Value** dialog appears.



- 3) Make the necessary changes. For more information, see [To add a registry key Startup item](#).
- 4) Click **OK**. The Startup view will be updated to reflect the changes.

Editing Startup Menu Items

- 1) Right-click the item with properties you wish to edit.
- 2) Select **Edit** from the pop-up menu. The **Edit File** dialog appears.
- 3) Make the necessary changes. For more information, see [Adding Start Menu Startup Items](#).
- 4) Click **OK**. The Startup view will be updated to reflect the changes.

NOTE

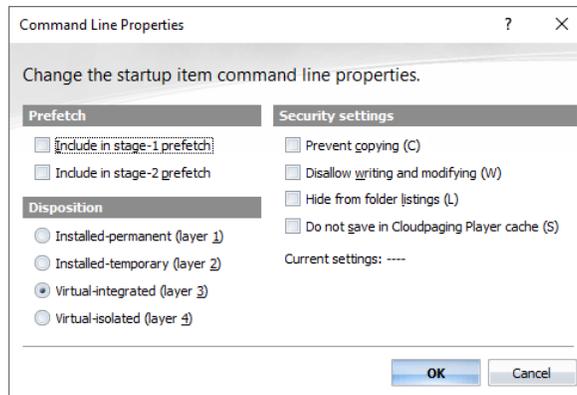
If the disposition of the registry key for a startup item is installed-permanent (layer 1) or installedtemporary (layer 2), but the corresponding executable file is set to virtual-integrated (layer 3) or virtual-isolated (layer 4), the host PC will not be able to see that executable file and consequently will not be able to launch the startup item. The startup item will be flagged in the workspace with the disposition in red to indicate this potential problem. For more information about conflict alerts, see [Conflict Color Coding](#).

Command Line File Properties of a Start Menu Item

Adding a Start Menu Startup item also adds the file to the Files view. You can edit the prefetch, disposition, and security settings from within the Startup view, rather than having to locate and edit the file from the Files view.

To edit the file properties of a Startup item:

- 1) Right-click the item with properties you wish to edit.
- 2) Select **Edit Command Line File Properties** from the pop-up menu. The **Command Line Properties** dialog appears.



- 3) Make the necessary changes. For more information, see [Editing File Properties](#).
- 4) Click **OK**. The Startup view will be updated to reflect the changes.

NOTE

If a startup item points to a shortcut (.lnk file) which has a disposition of installed-permanent (layer 1) or installed-temporary (layer 2), but the target executable file is set to virtual-integrated (layer 3) or virtual-isolated (layer 4), the item will be flagged with the disposition in red, because the shortcut will be physically present on the host PC, while the file to which it points is not. For more information about conflict alerts, see [Conflict Color Coding](#).

Deleting Startup Items

It is possible to delete any Startup item that was created during installation or that was added manually.

To delete a Startup item:

- 1) Right-click the Startup item that you wish to delete.
- 2) Select **Delete** from the pop-up menu.
- 3) Select **OK** when prompted for confirmation.

Excluding Startup Items

Excluded Startup items are not included when the appset is cloudified, but are still part of the project. This gives you the flexibility of including certain Startup items for one deployment of an appset and not for another. Excluding items also enables you to test an appset without a Startup item before permanently deleting it from the project.

To Exclude a Startup item:

- 1) Right-click the Startup item that you wish to exclude.
- 2) Select **Exclude** from the pop-up menu.
- 3) Excluded items are shown with their name and properties in italicized text.
- 4) If you wish to exclude another item, repeat steps 1-3.

Locating the File or Registry Entry of a Startup Item

If you need to view the Files View entry for a command file or the registry key for the selected startup item, **Cloudpaging Studio** will take you to the correct item directly from the Startup View.

Locating the command file in the Files View

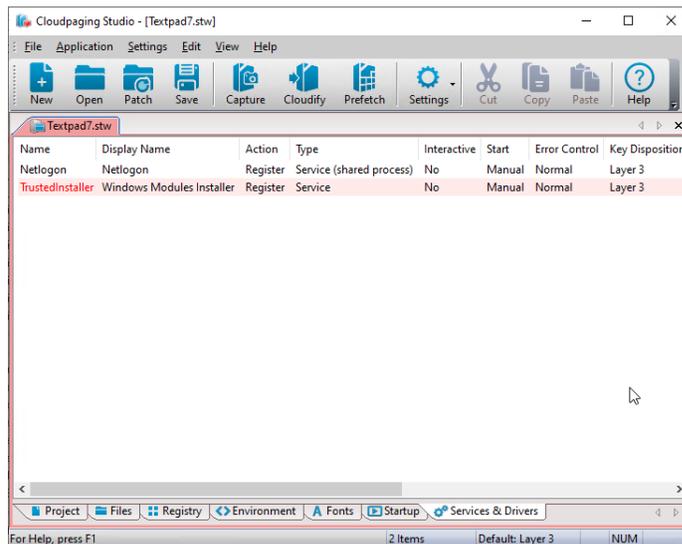
- 1) Right-click the startup item for which you wish to view the command file.
- 2) Click **Jump To** and select **Startup Command File**.
- 3) **Cloudpaging Studio** will switch to the Files View, select the correct folder and highlight the appropriate file.

Locating the registry value in the Registry View

- 1) Right-click the startup item for which you wish to view the registry value.
- 2) Click **Jump To** and select **Startup Value**.
- 3) **Cloudpaging Studio** will switch to the Registry View, select the correct key, and highlight the value.

Modifying Services and Drivers

Services installed during the capture process will be shown on the Services & Drivers view. You can manually add other services to a project. Most relevant commands can be accessed either from the **Edit** menu or by right-clicking on a service item.



NOTE

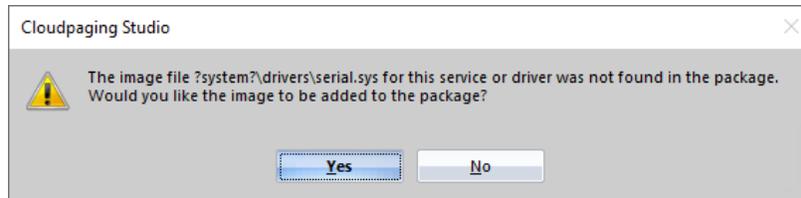
Most of what follows applies to both services and drivers since a driver is a special kind of service. So in the remainder of this section we will not distinguish between the two unless the described functionality or behavior is specific to one or the other.

Driver and services are shared components within Windows, which means that while the files and registry keys can be virtual, the running process will affect the entire system. Certain types of services or drivers must be ran at system startup and these, as well as any dependencies, may need to be made to layer-Removing such a service or driver may then require a system reboot by using a CAE.

Adding Services

To add a service to a project:

- 1) Right-click a blank area of the Services & Drivers view.
- 2) Select **Import Service** from the pop-up menu. The **Import Service** dialog appears.
- 3) Select the service from the list.
- 4) Click **OK**.
- 5) **Cloudpaging Studio** will prompt you to add the corresponding image file, if necessary.



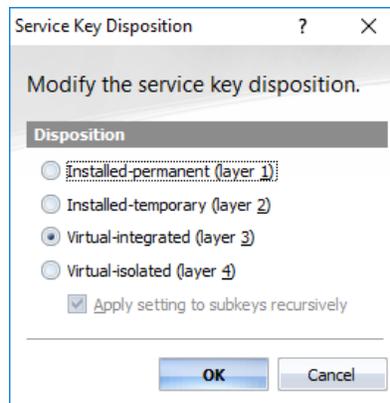
If you are certain the service or driver image file will be present on the Target PC when the application is virtualize, you can click **No**. Click **Yes** to add the image file to the project. It will appear in the Files view.

- 6) The Service & Drivers view will be updated to reflect the added service.

Editing the Registry Key Properties of a Service

To change the disposition of a registry key:

- 1) Right-click the desired item, and select **Service Key Disposition** from the pop-up menu.



The **Service Key Disposition** dialog appears.

- 2) Select the disposition. For more information about disposition settings, see [Changing the Disposition of a Registry Key](#).
- 3) The **Apply setting to subkeys recursively** setting will be checked automatically.
- 4) Click **OK**.

The Services & Drivers view will be updated to reflect the changed disposition setting for the selected item.

Editing the Image File Properties of a Service

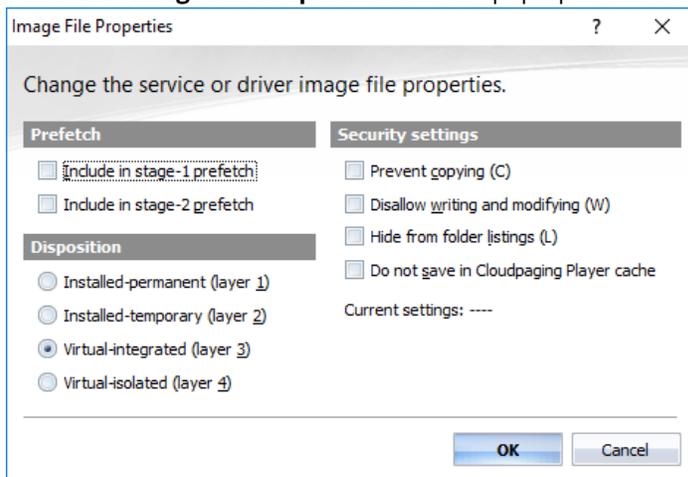
If the corresponding file for a Services & Drivers item has also been added to the Files view, you can edit the prefetch, disposition, and security settings from within the Services view, rather than having to locate and edit the file from the Files view.

NOTE

Any services in an appset on layer 3 are guaranteed to have all of their dependencies (e.g. .dll, .ini, .manifest, .cfg, etc. files) available to them when they need them. When setting services assets to physical layers (layer 1 or 2) however, you must be sure any of these dependencies are also layer 1 or 2. This is important if the services are meant to automatically start at Windows logon. Dependency Walker is a tool that helps identifying DLL files that an executable needs to run successfully.

To edit the file properties of a service:

- 1) Right-click the item with properties you wish to edit.
- 2) Select **Edit Image File Properties** from the pop-up menu. The **Image File Properties** dialog appears.



- 3) Make the necessary changes. For more information, see [Editing File Properties](#). Click **OK**. The Startup view will be updated to reflect the changes.

NOTE

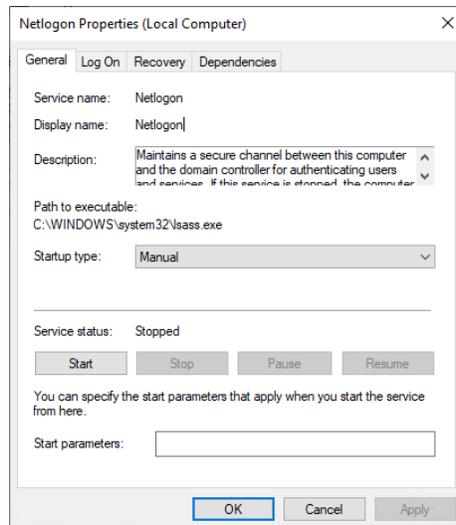
If there is a conflict involving one of the items, the relevant fields will appear in red. For details about the specific reason for this alert, see [Conflict Color Coding](#).

Editing Services

It is possible to directly edit the operating system properties dialog for any service that was created during installation or that was added manually. These are also accessible by opening Control Panel > Administrative Tools > Services, right-clicking on the name of a service, and selecting properties. **Cloudpaging Studio** is providing a shortcut to the properties dialog from within the application.

To edit a service:

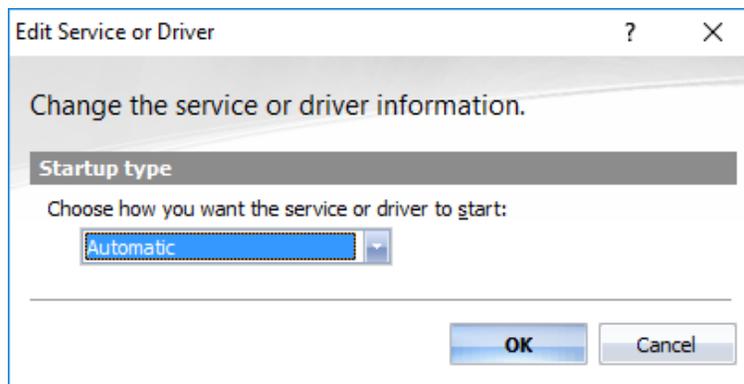
- 1) Right-click the service that you wish to edit.
- 2) Select **Edit** from the pop-up menu.



- 3) Windows will display the properties dialog for that service.
- 4) Select **OK** when finished to close the dialog, or **Cancel** if no changes are made.

To edit a driver:

- 1) Right-click the driver that you wish to edit.
- 2) Select **Edit** from the pop-up menu.
- 3) A dialog will display the properties for that driver
- 4) Select **boot loader**, **system**, **automatic**, **manual**, or **disabled** for the driver.



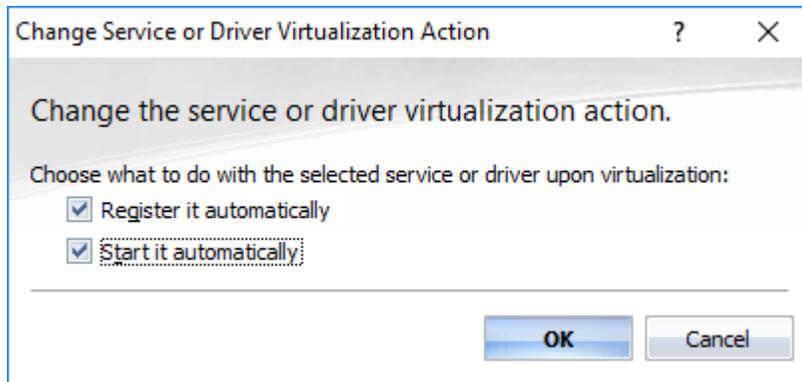
- 5) Select **OK** when finished to close the dialog, or **Cancel** if no changes are made.

Setting Virtualization Actions for a Service

The virtualization action determines how the selected service or driver is handled upon virtualization of the application on the Target PC.

To edit the virtualization actions for a service:

- 1) Right-click the service that you wish to edit.
- 2) Select **Change Virtualization Action** from the pop-up menu.
- 3) **Cloudpaging Studio** will display the **Change Service or Driver Virtualization Action** dialog for that service.



- 4) Check the desired settings. There are two options:
Register them automatically — Any services and drivers installed with the application will be registered with the Target PC upon virtualization.
Start them automatically – Registered services and drivers are started upon virtualization.
- 5) Select **OK** when finished to close the dialog, or **Cancel** if no changes are made.

Changing the Failure Action of a Service

Windows services will attempt to restart themselves if they crash for any reason. **Cloudpaging Studio** allows to you to disable this for some services that are part of an appset.

To change the failure action for a service:

- 1) Right-click the service for which you wish to change the failure action.
- 2) Select **Turn Failure Actions OFF** from the pop-up menu, if restart is enabled, or **Turn Failure Actions ON** from the pop-up menu, if restart is already disabled.
- 3) The option will be grayed out if the failure action cannot be changed for the selected service or driver.
- 4) If failure actions are ON for a service or driver, the name of that service will be shown in red on the Services & Drivers view.

Deleting Services

It is possible to delete any service that was created during installation or that was added manually.

To delete a service:

- 1) Right-click the service that you wish to delete.
- 2) Select **Delete** from the pop-up menu.
- 3) Select **OK** when prompted for confirmation.

Excluding Services

Excluded services are not included when the appset is cloudified, but are still part of the project. This gives you the flexibility of including certain services for one deployment of an appset and not for another. Excluding services also enables you to test an appset without a services before permanently deleting it from the project.

To exclude a service:

- 1) Right-click the service that you wish to exclude.
- 2) Select **Exclude** from the pop-up menu.
- 3) Excluded services are shown with their name and properties in italicized text.
- 4) If you wish to exclude another service, repeat steps 1-3.

Locating the Image File or Registry Key of a Service

If you need to view the Files View entry for an Image File or the Registry Key for the selected service or driver, **Cloudpaging Studio** will take you to the correct asset directly from the Services & Drivers View.

Locating the image file in the Files View

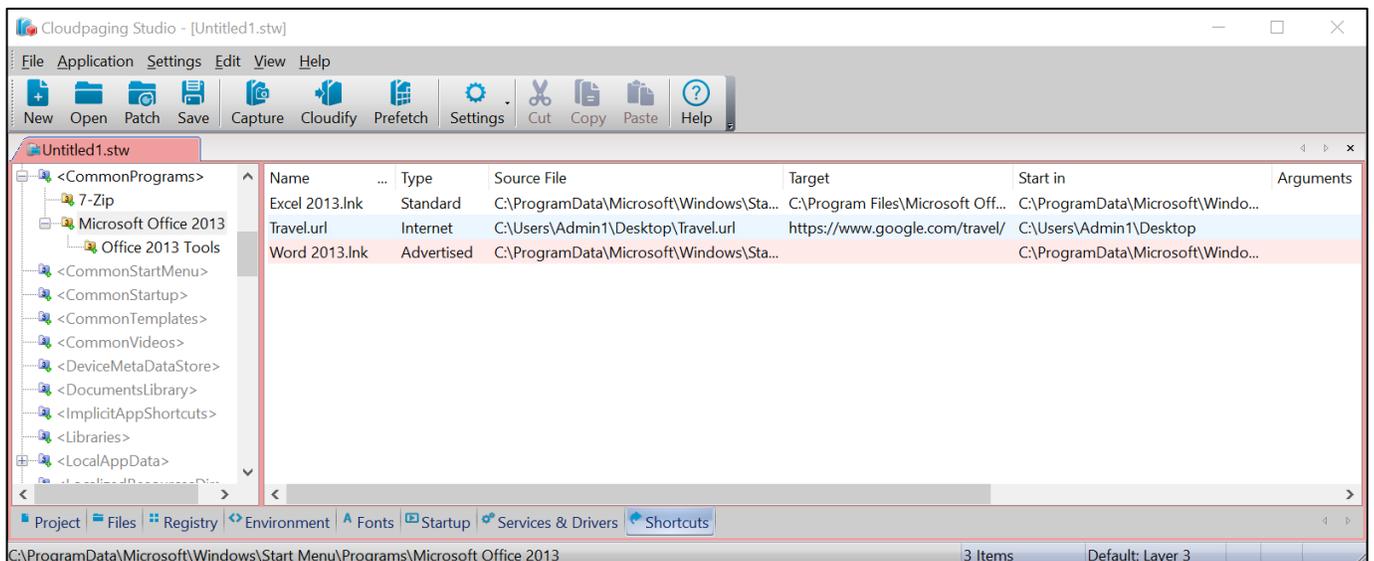
- 1) Right-click the service for which you wish to view the image file.
- 2) Click **Jump To** and select **Service File**.
- 3) **Cloudpaging Studio** will switch to the Files View, select the correct folder and highlight the appropriate file.

Locating the registry key in the Registry View

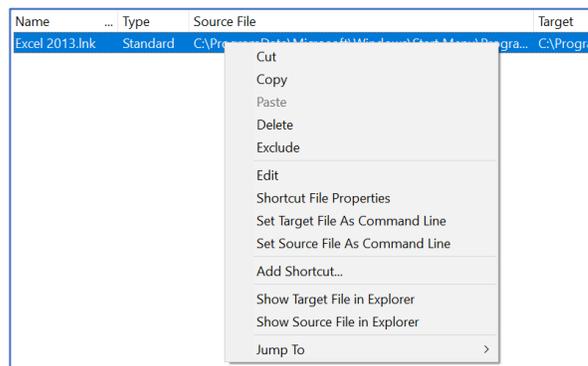
- 1) Right-click the service for which you wish to view the registry key.
- 2) Click **Jump To** and select **Service Key**.
- 3) **Cloudpaging Studio** will switch to the Registry View and select the correct key.

Modifying Shortcuts

The Shortcuts view is accessed by selecting **Shortcuts** from the **View** menu, or by selecting the **Shortcuts** view as shown below.



This view displays shortcuts that have been created either during installation or manually added.

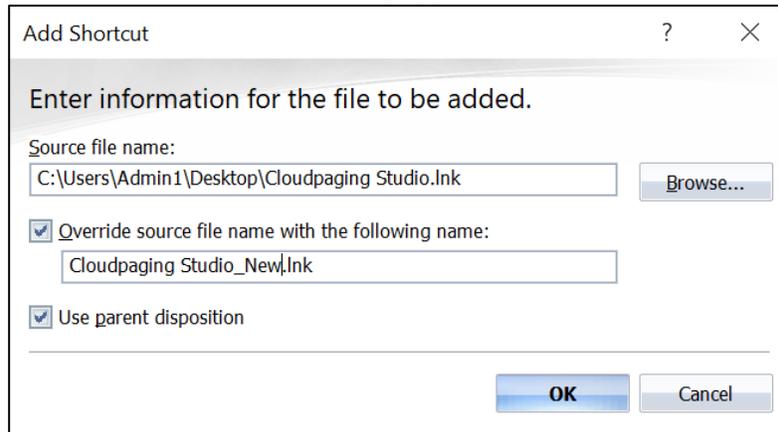


Most relevant commands can be accessed either from the **Edit** menu or by right-clicking on a shortcut. Shortcuts shown in a red color indicates that the target is missing in both the project and on the system, such as advertised shortcuts that have not been installed. Shortcuts shown in an orange color indicates that the target is present on the system but not in the project. Please refer to [Understanding Color Coding in Views](#) for additional information.

Adding Shortcuts

To add a shortcut to a project:

- 1) Right-click a blank area of the Shortcuts view.
- 2) Select **Add Shortcut...** and the **Add Shortcut** dialog appears.

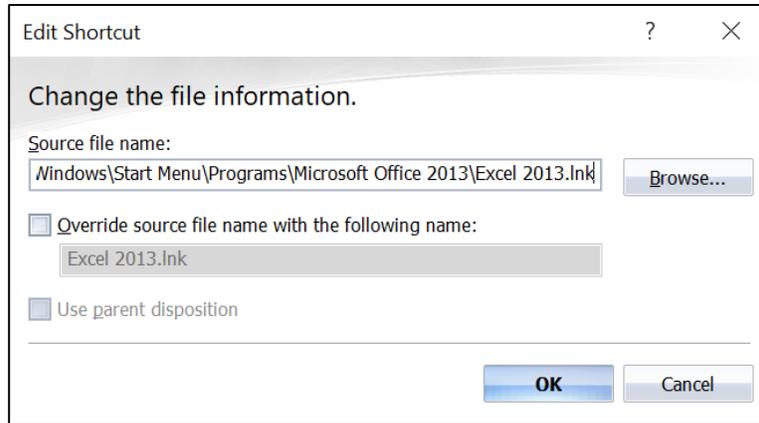


- 3) Enter the **Source file name** to the shortcut or **Browse** to find the shortcut on the system.
- 4) If the name needs to be different, then select the **Override source file name with the following name** and enter the new name.
- 5) If the disposition layer should be inherited from the parent folder, then check the **Use parent disposition** checkbox.
- 6) Click **OK**.

Editing Shortcuts

To edit a shortcut:

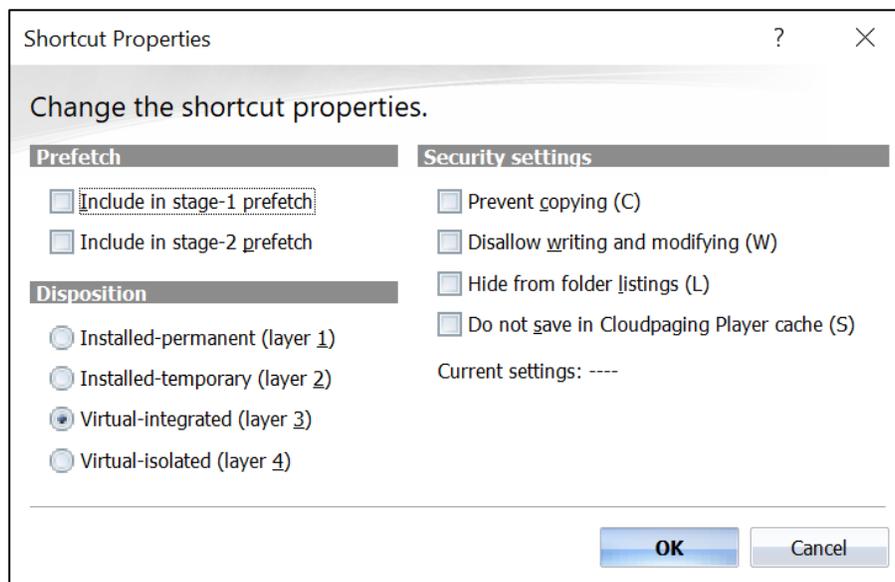
- 1) From the Shortcuts view, right-click the shortcut that you wish to edit.
- 2) Select **Edit** from the pop-up menu. The **Edit Shortcut** dialog appears.



- 3) Make the necessary changes. For more information, see [Adding Shortcuts](#).
- 4) Click **OK**. The Shortcuts view will be updated to reflect your changes.

To edit the file properties of a shortcut:

- 1) Right-click the shortcut with properties you wish to edit.
- 2) Select **Shortcut File Properties** from the pop-up menu. The **Shortcut Properties** dialog appears.



- 3) Make the necessary changes. For more information, see [Editing File Properties](#).
- 4) Click **OK**.

Setting the Command Line

Most applications will place a primary shortcut on the desktop or start menu to access an application. You can set the command line to be that of a shortcut or the target of a shortcut.

Name	Type	Source File	Target	Start in	Arguments
Excel 2013.Ink	Standard	C:\ProgramData\Microsoft\Windows...	C:\Program Files\Microsoft ...	C:\ProgramData\Microsoft\Win...	
Travel.url	Internet	C:\Users\Admin1\Desktop\Travel.url	https://www.google.com/travel/	C:\Users\Admin1\Desktop	
Word 2013.Ink	Advertised	C:\ProgramData\Microsoft\Windows\Sta...		C:\ProgramData\Microsoft\Windo...	

To set a shortcut's target file as a command line

- 1) Right-click the desired shortcut.
- 2) Click **Set Target File as Command Line**.
- 3) **Cloudpacing Studio** will bold the shortcut to indicate it is the command line.

Setting the shortcut's target as the Command Line will also set the Working Folder to the "Start in" folder location. If the "Start in" location is empty, then the folder of the target file will be used.

To set a shortcut as a command line

- 1) Right-click the desired shortcut.
- 2) Click **Set Source File as Command Line**.
- 3) **Cloudpacing Studio** will bold the shortcut to indicate it is the command line.

Setting the shortcut as the Command Line will also set the Working Folder to folder location of the shortcut.

Deleting Shortcuts

It is possible to delete any shortcut that was created during installation or that was added manually.

To delete a shortcut:

- 1) Right-click the shortcut that you wish to delete.
- 2) Select **Delete** from the pop-up menu.
- 3) Select **OK** when prompted for confirmation.

Excluding Shortcuts

Excluded shortcuts are not included when the appset is cloudified, but are still part of the project. This gives you the flexibility of including certain shortcuts for one deployment of an appset and not for another. Excluding shortcuts also enables you to test an appset without a shortcut before permanently deleting it from the project.

To exclude a shortcut:

- 1) Right-click a shortcut that you wish to exclude.
- 2) Select **Exclude** from the pop-up menu.
- 3) Excluded shortcuts are shown with their name and properties in italicized text.
- 4) If you wish to exclude another shortcut, repeat steps 1-3.

Locating the source or target for a Shortcut

If you need to view the source or target for selected shortcut, **Cloudpacing Studio** will take you to the correct asset directly from the Shortcuts View. In addition, if you need to view the source or target for selected shortcut, **Cloudpacing Studio** will take you to the shortcut in Explorer.

Name	Type	Source File	Target
Excel 2013.lnk	Standard	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Microsoft Office 2013\Excel 2013.lnk	C:\Program
travel.url	Internet	C:\Users\Admin1\Desktop\travel.url	https://www
Word 2013.lnk	Advertised	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Microsoft Office 2013\Word 2013.lnk	

Locating the shortcut target file in the Files View

- 1) Right-click the shortcut for which you wish to view the target file.
- 2) Click **Jump To** and select **Target File**.
- 3) **Cloudpaging Studio** will switch to the Files View, select the correct folder and highlight the appropriate file.

Locating the shortcut source file in the Files View

- 1) Right-click the shortcut for which you wish to view the source file.
- 2) Click **Jump To** and select **Source File**.
- 3) **Cloudpaging Studio** will switch to the Files View, select the correct folder and highlight the appropriate file.

Locating the shortcut target file in Explorer

- 1) Right-click the shortcut for which you wish to find the target.
- 2) Click **Show Target File in Explorer**.
- 3) The corresponding shortcut target folder will be opened.

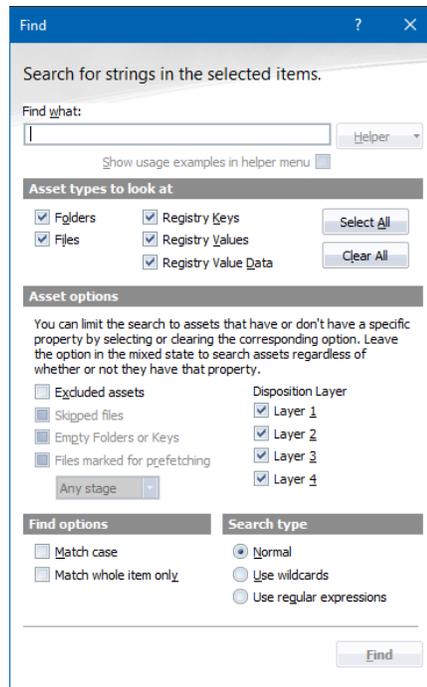
Locating the shortcut source file in Explorer

- 1) Right-click the shortcut for which you wish to find source.
- 2) Click **Show Source File in Explorer**.
- 3) The corresponding shortcut folder will be opened.

Searching for Assets

Finding Assets

The **Find** feature can be used to locate folders, files, registry keys and values that match specific criteria.



To find assets:

- 1) From any view, select the **Find** menu under the **View** menu (or press F3). The **Find** dialog appears.
- 2) Select the type of search: **Normal**, **Using wildcards**, or **Using regular expressions** (see [About Search Types](#)).
- 3) Choose the areas to search under **Look at**:
 - **Folders** - Names of the directories seen in the File view.
 - **Files** - Names of the files found in folders seen in the File view.
 - **Registry Keys** - Names of the registry keys seen in the Registry view.
 - **Registry Values** - Strings contained within the registry keys seen in the Registry view.
 - **Registry Value Data** - The strings or digits assigned to Registry values.
- 4) Check or uncheck the desired **Asset Options**:

These check boxes start up in a mixed state, neither checked nor empty, meaning that search will not consider the selected option or options. Not all options are available for all combinations of asset types. For example, skipped files applies to files and files only. If you select any asset type other than files, even if you also select files, skipped files will not be available during that search.

 - **Excluded assets** - Checking this option will restrict the search to assets marked as Excluded in any view. Unchecking this option will restrict the search to assets not marked as Excluded.
 - **Skipped files**— These files were missing or inaccessible during packaging. These files are also automatically marked as Excluded. Checking this option will restrict the search to skipped files. Unchecking this option will restrict the search to files that were not skipped.
 - **Empty Folders or Keys** - Checking this option will restrict the search to folders that contain no files or registry keys with no values. Unchecking this option will restrict the search to nonempty assets.
 - **Files marked for prefetching** - Check this option will restrict the search to files marked for prefetching during the selected stage or stages. Unchecking this option will restrict the search to files not marked for prefetch.
- 5) Check or uncheck the desired **Find Options**:

- **Match case** – With this checked, “Network” will not match “NETWORK.”
- **Match whole item only** – With this checked, “Photoshop” will not match “Adobe Photoshop.”

NOTE

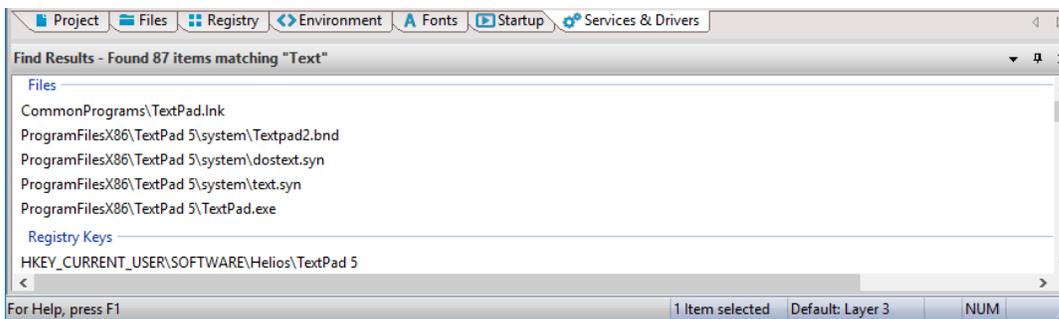
The Search feature searches only the File and Registry views. The other views are special cases of these views. For example, Startup items also appear in either the File or Registry views. Search will locate those items within those two views.

- 6) Enter a search string in the **Find what** field.
- 7) Click the **Find** button.

The results of the search will be displayed in the **Find Results** window (see [The Find Results Window](#) below) located in the lower half of the **Cloudpaging Studio** window by default.

The Find Results Window

The results of a search are displayed in the **Find Results** window and grouped by asset type. The window is located in the lower half of the **Cloudpaging Studio** window by default but you can move it around by dragging its title bar to another edge of the main window or anywhere else to keep it floating.



From within the window, it is possible to jump to a specific asset in its associated view. Additionally, the **Find Results** window allows multiple assets to be selected and **Properties** can be changed for all selected assets at once as long as they are of the same asset type.

It is also possible to selectively remove results from the window if it is too crowded with results of no interest.

To jump to a specific asset:

- Double-click the asset you wish to jump to.

- OR -

- Right-click the asset you wish to jump to and select **Jump To**.

To change properties:

- 1) Select the assets you wish to change.

NOTE

Use **Ctrl-click** to select multiple assets. To select a range of consecutive assets, click the first asset in the range then drag the mouse cursor to the last asset. To select multiple ranges, press **Ctrl** while selecting each range.

- 2) Right-click the selection and select **Edit**.

- 3) Update the properties and apply the changes.

For example, to set security on all EXE files, find ***.exe** and select all of the files. Then select **Edit** and change the security options for the selected files.

To remove assets from the results list:

- 1) Click the assets you wish to change.

NOTE

Use Ctrl-click to select multiple assets. To select a range of consecutive assets, click the first asset in the range then drag the mouse cursor to the last asset. To select multiple ranges, press Ctrl while selecting each range.

- 2) Right-click the selection and select **Remove from list**.

To clear the results list:

- To remove all results from the list, right-click any asset and select **Clear list**.

About Search Types

Cloudpaging Studio has three types of searches:

- **Normal** – normal searches look for an exact match on a string of characters.
- **Use wildcards** (see [Using Wildcards](#) below).
- **Use regular expressions** (see [Using Regular Expressions](#) below).

Using Wildcards

These search queries use the standard wildcard characters (* or ?) to help find patterns. An asterisk will search for any number of characters and the question mark will match 0 to 1 characters.

Using Regular Expressions

Regular expressions are a powerful tool for pattern matching. They are capable, for example, of finding a string of exactly six characters, beginning with an uppercase letter, followed by four lowercase letters and a single digit, and only at the beginning of a longer string. Regular expressions enable you to search with a high degree of precision and maximum flexibility.

Because regular expressions are an advanced topic beyond the scope of this guide, they are recommended for advanced users. The **Helper** menu next to the search box will get you started, but if you want to learn more, a good place to look would be [the tutorial at regular-expressions.info](#).

Globally Changing Asset Dispositions

This tool enables you to change the disposition layer for all assets of a specific type.

Important

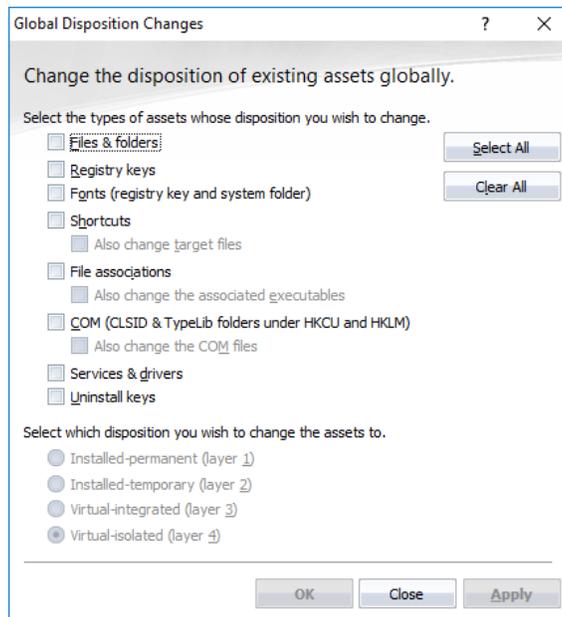
This is an extremely powerful feature that will make changes to every file, registry key, font, or driver. You should save your project before using this tool.

NOTE

The Global Disposition Settings dialog partially takes the place of the Virtual Settings option in previous version of Cloudpaging Studio. The other part of this functionality is handled by the Sandboxing screen in the Settings dialog. See [Sandboxing Settings](#).

To Globally Change Dispositions:

- 1) Click the Edit menu and select Global Disposition Settings. The Global Disposition Settings dialog appears.



- 2) Check the types of asset that will be affected:
 - **Files** – This will affect all files under the Files view.
 - **Registry Keys** – This will affect all registry keys in the Registry view.
 - **Fonts** — This will affect all fonts in the Fonts view. Note that Fonts are also shown in the Files view, so changes to Files also affects fonts.
 - **Short Cuts** — This will affect shortcuts on the Desktop or Start Menu.
 - **File Associations** — This will affect any file type associations created by the cloudified application. You also have the option of checking a second box to change the disposition of the executable files associated with the file types.
 - **COM** — This will affect registry keys for COM-based libraries. Check the second box to also change the disposition of the related image files.
 - **Services and Drivers** — This will affect registry keys and image files for assets in the Services & Drivers view.
 - **Uninstall Keys** — This will affect any entries for the cloudified application on the Windows Remove Program dialog.
- 3) Select the new disposition layer.
- 4) Click **OK** or **Apply**. The Apply button leaves the Global Disposition Settings open, so you can make more changes.

Understanding Color Coding in Views

If you see an asset displayed in color in one of the workspace views, that indicates something about the status of that asset. If you hover your mouse over the asset, you should see brief description of the conflict in a tool tip message.

The following table shows how assets are color-coded in Views.

Color/Attribute	Asset Type	Meaning
Black	All Assets	Normal
Grey	Template Folders	Empty
Grey Italic	Files Root Folders Sub-folders Registry Keys Registry Values	Excluded
Blue	Files Folders Registry Keys Registry Values	New (After capture) ¹
Magenta	Files Folders Registry Keys Registry Values	Changed (After capture) ²
Red	Files Folders Registry Keys Registry Values	Skipped during cloudifying.
	All Assets	Missing asset or conflict. ³
Orange	Shortcuts	Target exists on system but missing in appset.
Bold	Folder	Working Folder (or ancestor).
	File Shortcut	Command Line

- 1) This does not apply to manually added assets.
- 2) This does not apply to manually changed assets.
- 3) See the table below details about the probable cause of the conflict.

Conflict Color Coding

The following table details the various conditions that will result in a conflict (red) condition.

View(s)	Color-Coded Field	Meaning
Font StartUp Services & Drivers	File Command Line	File associated with asset is not in the project. Note: This may be okay if the file is known to always be on the end user system and at the same location (e.g. notepad.exe) Note (exception): If the file is in the project but excluded, it is shown in grey italic (as indicated in the previous table). This is the only case where a non-red color indicates a potential conflict.
Font	Action	Fonts are set to be registered, but the font will not be visible to the system because the Fonts key is set to layer 4.
	Action + Disposition	Fonts are set to be registered, but the font will not be visible to the system because the font file is set to layer 4.
Startup (Registry)	Disposition	Registry Run (or RunOnce) key asset cannot be started by the system because the key is set to layer 1 or 2, but its target command line is set to layer 3 or 4 and will not be visible to the system.
StartUp (Files)	Name + Disposition	One of these two conditions exists: <ul style="list-style-type: none"> • Shortcut file does not exist on the system. • Startup shortcut cannot be started by the system because it is set to layer 1 or 2, but its target command line is set to layer 3 or 4 and will not be visible to the system.
Services and Drivers	Name	Service is set to restart or reboot automatically upon failure. This will cause the service to always restart when the application is shut down from the Player, which will put it back into the Running state. Toggling this service's Failure Actions to OFF will fix this issue.
	Action	Action cannot be performed on a service whose type is not specified.
	Type	Service type is not specified.
	Interactive	Interactive setting is not specified.
	Start	Start type is not specified.
	Error Control	Error control type is not specified.
	Image Disposition	Driver is set to start on boot or system start, but its service key and/or image file is set to layer 3 or 4 and will not be visible to the system at that time.
Both Key and Image Dispositions	Image file layer is greater than service key layer, which can yield inconsistent behavior. Note: One exception -- it is okay for the service key to be at layer 3 and the image file at layer 4.	

Chapter 8: Modifying an Appset

Appsets can be modified by editing either the Application Installation Blueprint file (.aib) and Token files (.tok). The AIB file consists of metadata about the appset's file and registry settings, and the Token file consists of license key data and information provided by the software publisher.

The purpose of modifying the AIB or TOK file is to allow for changing security settings, disposition, prefetch settings, file attributes, as well as file names and locations. Modifying AIB files is particularly useful when modifying appset settings during Phase 4 of the cloudifying process (Testing and Optimizing the Appset). Modifying the AIB file saves the extra time that would be spent re-cloudifying the appset after each testing iteration.

NOTE

New files can not be added to an appset (i.e. AIB or TOK file). A new appset must be created or an appset must be patched if new files are required.

Additionally, the EULA or Icon file for that appset can also be modified by following the [Opening an Appset file](#) and [Saving an Appset File](#) instructions.

Important

All modifications made by editing the TOK and/or AIB file are not saved in the original STW file. Thus when patching an appset, any of these modifications would be lost on new patch revisions. To avoid this inconsistency, you should make modifications for testing only and reapply final changes to the original STW.

The following are explained in this chapter:

- Opening an Appset file
- Editing an AIB file
- Editing Token files
- Saving an Appset File

Opening an Appset file

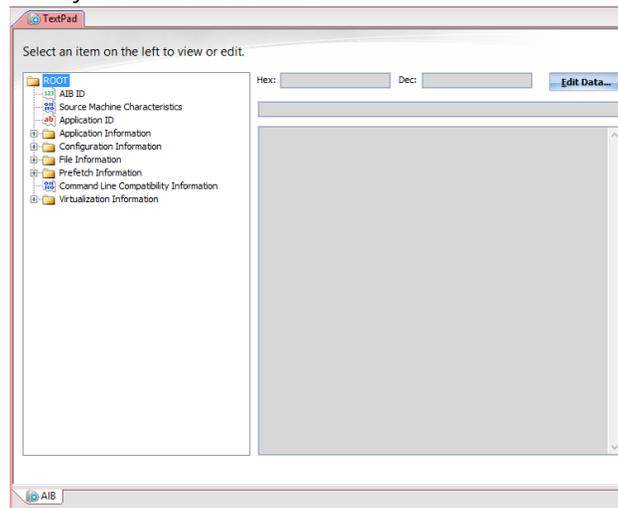
Before an appset can be modified, it must first be extracted from the appset or **.stp** file.

NOTE

You can open a Read-Only token file for viewing, but it cannot be edited. In order to edit a token file, you must have read and write permission for the token file.

To open an appset's Token or AIB file:

- 1) Unzip the **.stp** file by renaming it as a **.zip** file, right-click it, and select **Extract Here**. For example, rename "test.stp" to "test.zip" and then extract the contents.
The contents of the **.stp** file (**.stc**, **.aib**, **.tok**, etc.) is extracted to the folder you selected.
- 2) Double-click the AIB or Token file.
The contents of the AIB or Token file appears on the left pane of the **Cloudpaging Studio** UI.
You are now ready to modify the AIB or Token file.



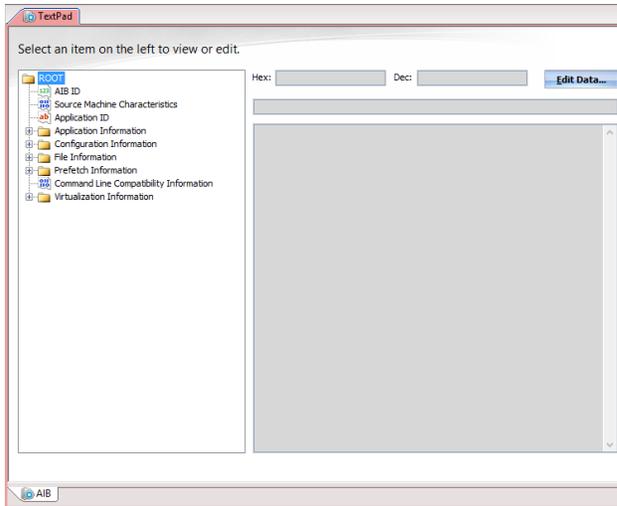
Editing an AIB file

This section covers the modifications that can be made to the AIB file.

Modifying File Settings

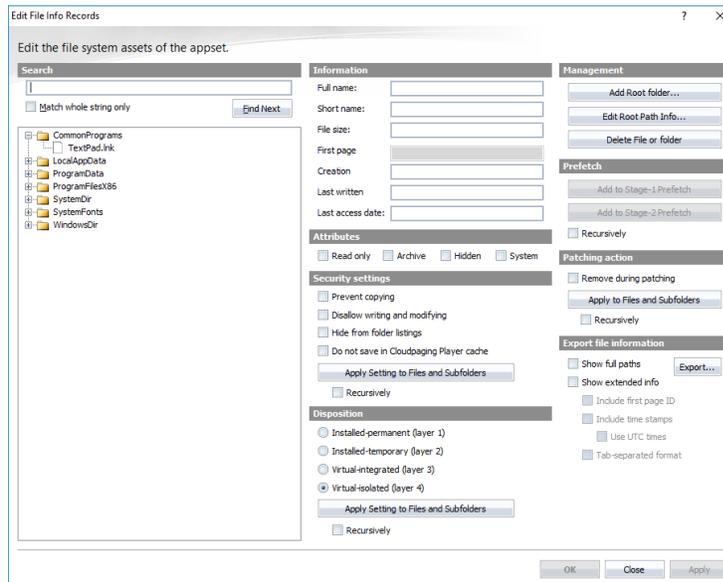
To modify file settings of an AIB file:

- 1) Open the AIB file. See [Opening an Appset file](#) for instructions.
The AIB view appears.



- 2) Expand File Information, and double-click File Records.

The Edit File Info Records dialog appears.



- 3) Select a file or folder and make the appropriate setting changes. Most of these fields and options are explained earlier in this manual. Refer to references as cited.

- Information:
 - **Full name** - The complete name of the selected assets.
 - **Short name** - The short name of the selected assets.

Important

*The **Short Name** must be identical to that of the .stc file associated with the Token File. If this field is incorrect, then the application will not run properly and the **Cloudpaging Player** will display an error message.*

- File size, First Page Id, Creation date, Last written date, Last access date - These fields are all auto-generated.
 - Attributes: Read only, Archive, Hidden, System - These options are the same as those found on the Windows file or folder Properties dialog.
 - Security Settings: Prevent copying, Disallow writing and modifying, Hide from folder listings, Do not save in Cloudpaging Player cache (see [Editing Folder Properties](#)).
 - Disposition: Installed-permanent (layer 1), Installed-temporary (layer 2), Virtual-integrated (layer 3), Virtual-isolated (layer 4). For more information see [Editing Folder Properties](#).
 - Management (buttons): Add New File, Add New Subdirectory, Edit Root Path Info, Delete File or Directory - These functions are as labeled.
 - Prefetch:
 - Add Directory to Stage-1 Prefetch - Adds the current folder to the Stage-1 prefetch and adds Stage-1 prefetch to the appset, if necessary.
 - Add Directory to Stage-2 Prefetch - Adds the current folder to the Stage-2 prefetch and adds Stage-2 prefetch to the appset, if necessary.
- 4) Click **OK**.

Exporting File Information

Cloudpaging Studio can output a text file that lists every folder and file shown in this view, along with the option of showing attributes and timestamps.

Options

- **Show Full Paths** — If checked, the output file will contain the complete file path for each file. If not, the file will show a hierarchical tree view.
- **Show Extended Info** — If checked, the output file will contain attributes. If not, it will only contain the file/folder names and paths. If this is checked, the following options are available:
 - **Include First Page ID** — If checked, the file will contain the page ID for each file.
 - **Include time stamps** — If checked, the file will contain the creation, last modification, and last access timestamps for each file.
 - **Use UTC times** — If checked, the timestamps will be written in Greenwich Meridian or UTC time, rather than local times. This option is not available if timestamps are disabled.
 - **Tab-separated format** — If checked, the file will be output in a tab-delimited format for importing into spreadsheet programs such as Microsoft Excel. If not, it will be formatted for printing or viewing.
- **Export** — Click this button to name and save the file.

Searching for File Information

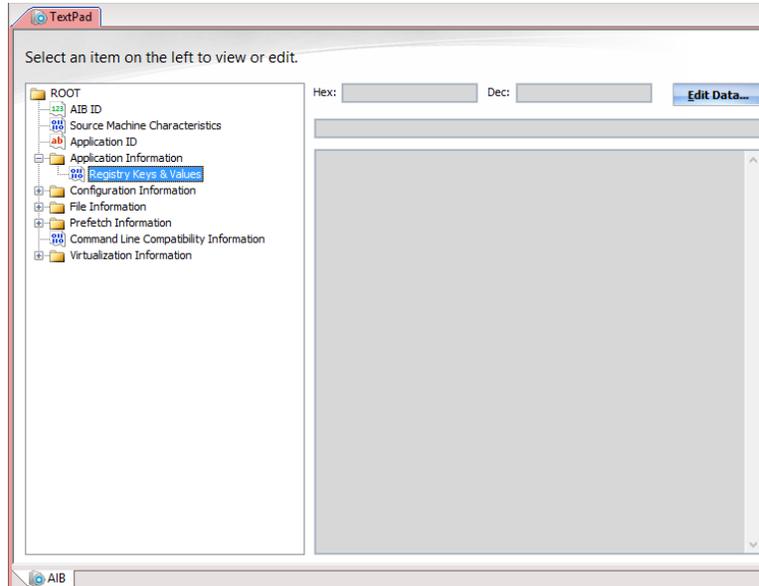
To locate a specific file:

- 1) Type a full or partial file or folder name into the Search box at the top of the **Edit File Info Records** dialog.
- 2) Click **Find Next**.
- 3) If a file or folder name matches the search, the first matching file or folder will be highlighted and its information displayed.
- 4) If this is not the correct asset, click **Find Next** until the desired file or folder is displayed.

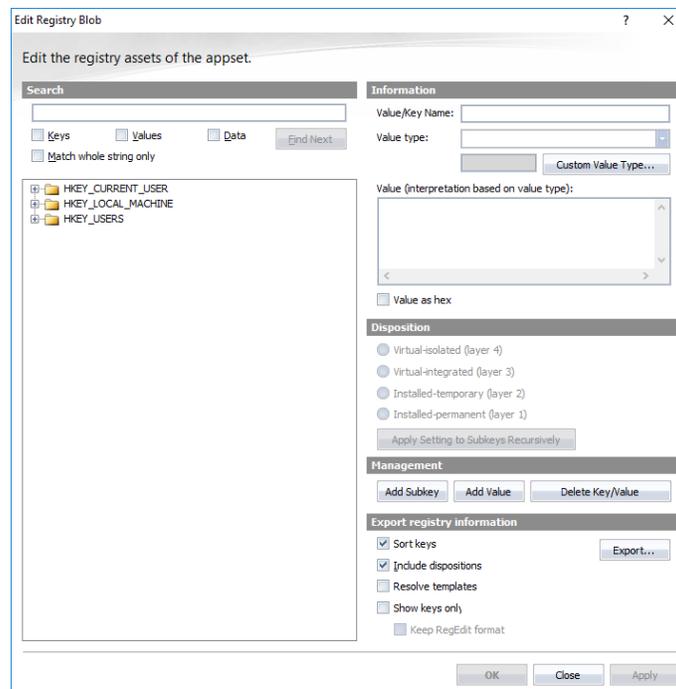
Modifying Registry Settings

To modify the registry settings of an AIB file:

- 1) Open the AIB file. See [Opening an Appset file](#) for instructions.
The AIB view appears.

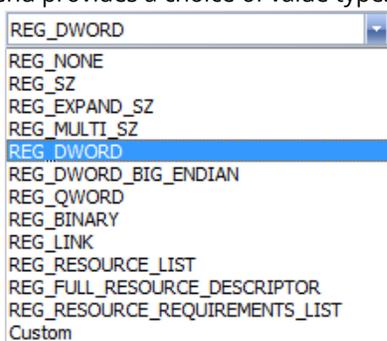


- 2) Expand Application Information, and double-click Registry Keys & Values. The Edit Registry Blob dialog appears.



- 3) Select a key or value and make the appropriate setting changes. Some fields and options are explained earlier in this manual. Refer to references as cited.
 - Information:

- **Value/Key Name** - The name of the registry value or key.
- **Value type** - The drop-down menu provides a choice of value types.



- **Value** - Auto-generated.
- **Value as hex** - Select this option to display the registry value as hexadecimal.
 - Disposition: **Installed-permanent (layer 1), Installed-temporary (layer 2), Virtual-integrated (layer 3), Virtual-isolated (layer 4)**. For more information, see [Editing Folder Properties](#).
 - Management (buttons): **Add Subkey, Add Value, Delete Key/Value**. These functions are as labeled.

Exporting Registry Information

Cloudpaging Studio can output a file, in RegEdit format, including all values, of every registry key in the project.

Options

- **Sort Keys** — If this is checked, the registry keys are written to the file sort alphabetically. If not, the keys are listed in the order found in the registry blob.
- **Show Keys Only** — If this is checked, the output file will be a flat listing of registry keys, one per line, without the values. If you check **Keep RegEdit Format**, each registry key will be in the form of a section header from a registry file, but without the values. In effect, **Cloudpaging Studio** will save a “blank” registry file.

Finding Registry Information

Cloudpaging Studio can locate individual keys and values using the search box at the top of the **Edit Registry Blob** dialog.

To locate a specific key or value:

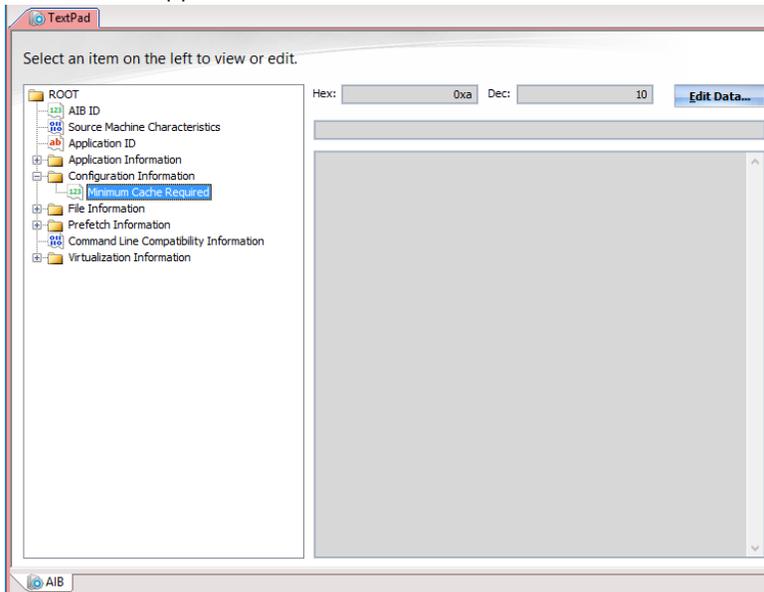
- 1) Type the full or partial name into the Search box at the top of the **Edit File Info Records** dialog.
- 2) Check the boxes to indicate the type of registry asset that you want to search: **Keys, Values, and Data**.
- 3) Click **Find Next**.
- 4) If a registry asset matches the search, the first matching asset will be highlighted and its information displayed.
- 5) If this is not the correct asset, click **Find Next** until the desired asset is displayed.

Modifying Minimum Cache Allocation

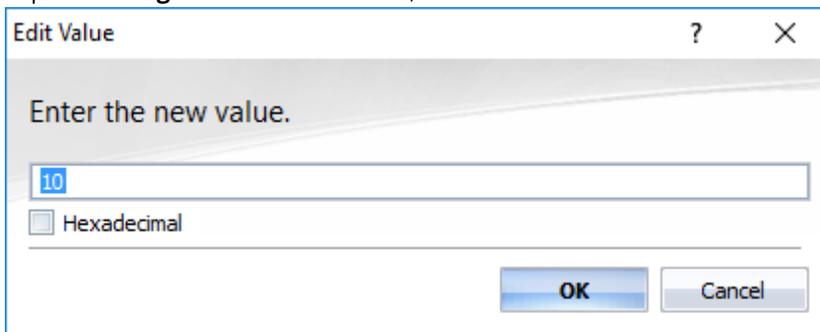
To modify the minimum cache allocation settings of an AIB file:

- 1) Open the AIB file. See [Opening an Appset file](#) for instructions.

The AIB view appears.



- 2) Expand **Configuration Information**, and double-click **Minimum Cache Required**

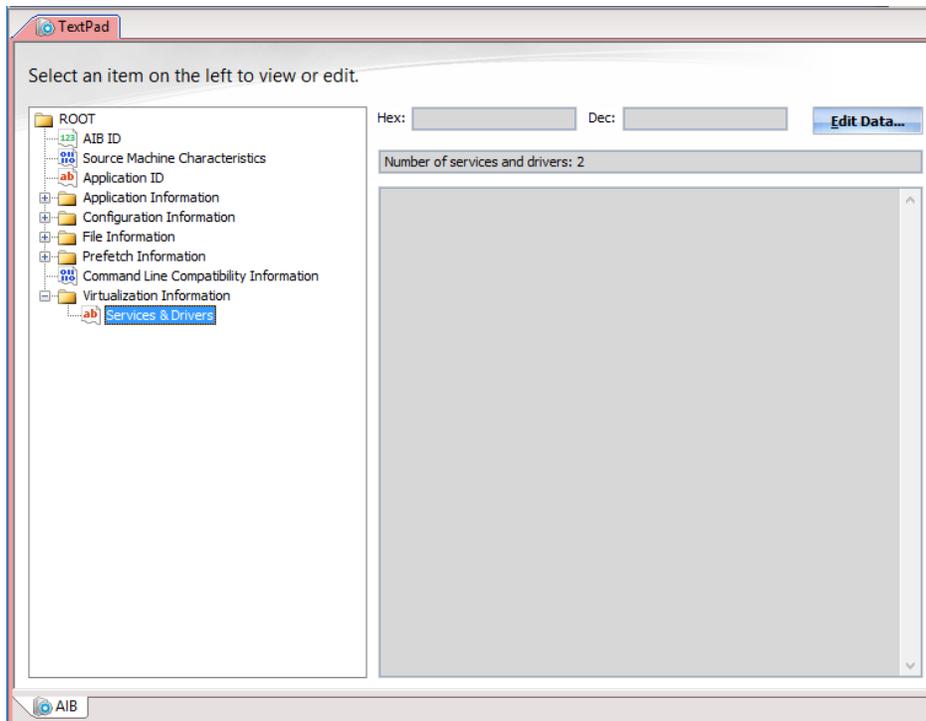


- 3) Enter the new value of 5 to 100, and click **OK**.

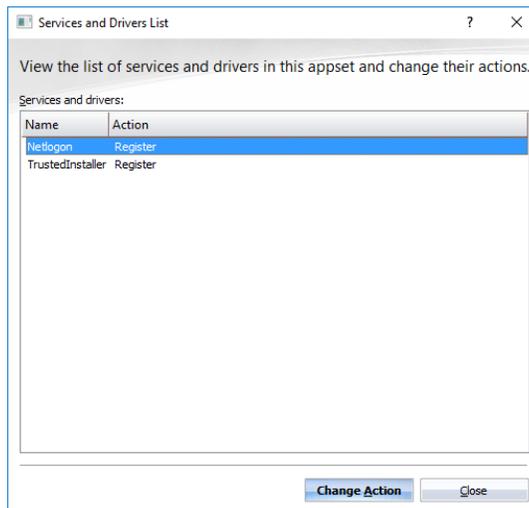
Editing Virtualization Information

To modify the virtualization settings of an AIB file:

- 1) Open the AIB file. See [Opening an Appset file](#) for instructions. The AIB view appears.



- 2) Expand **Virtualization Information**, and double-click **Services & Drivers**. **Cloudpaging Studio** displays the Services and Drivers List.



- 3) Select the desired service or driver and click **Change Action**. **Cloudpaging Player** displays the Change Service or Driver Virtualization Action dialog.
- 4) Select how you want **Cloudpaging Player** to handle the selected service and driver by checking or unchecking the boxes. For more information on these settings, see [Virtualization Settings](#).

Adding Prefetch Files

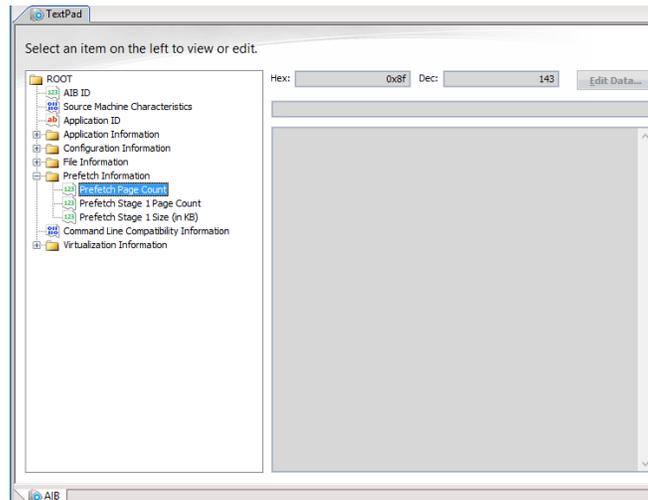
From the AIB editor, it's possible to add prefetch files that were captured by **Cloudpaging Player**. See [To capture and add a prefetch](#): for the steps to follow. The procedure is identical.

Viewing Prefetch Information

To view the prefetch statistics:

The AIB file can show you the size of your Prefetch pages.

- 1) Open the AIB file. See [Opening an Appset file](#) for instructions.
The AIB view appears.



- 2) Select any of the following to see the number of pages of each type, along with their size. Each page can be up to 32 kilobytes, but it could be smaller due to compression.
 - **Prefetch Page Count** – The total number of prefetch pages for all stages.

If there are any assets assigned to a prefetch stage, the following will appear:

- **Prefetch Stage 1 Page Count** – The number of stage-1 prefetch pages.
- **Prefetch Stage 1 Size (in KB)** – The size in kilobytes of the stage-1 prefetch pages.

If there are any pages assigned to prefetch stage 2, the following will appear:

- **Prefetch Stage 2 Page Count** – The number of prefetch pages loaded after the application has launched (stage-2).
- **Prefetch Stage 2 Size (in KB)**– The size in kilobytes of the prefetch pages loaded after the application has launched (stage-2).

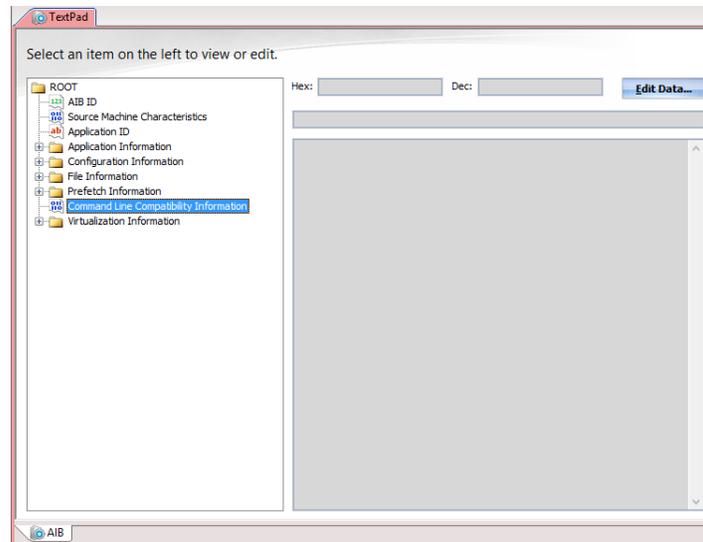
If there are no stage 1 prefetch pages but there at least one asset assigned to stage 2, the stage 1 page count and size will appear, but both values will be zero.

If there are files assigned to more than two prefetch stages in the cloudified application, those additional prefetch stages will also be listed.

Editing Compatibility Settings

To modify the Compatibility Settings in an AIB file:

- 1) Open the AIB file. See [Opening an Appset file](#) for instructions.
The AIB view appears.



- 2) Double-click Command Line Compatibility Information. Cloudpaging Studio displays the Compatibility Settings dialog.
- 3) Edit the desired settings. For more information, see [Compatibility Settings options](#).

Editing Token files

This section covers the modifications that can be made to the TOK file.

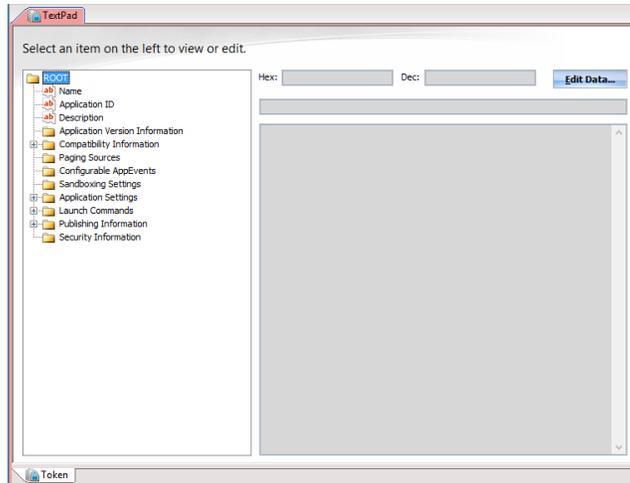
Modifying an Application Value

The Token values for the following can be changed:

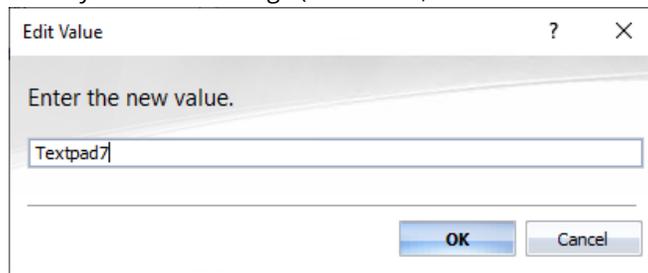
- Full name
- Short name
- Description
- Minimum Client Version Required
- System Requirements
- Reboot Flag
- Launch command line
- Working Directory
- Publisher Name

To change the value of a Token file:

- 1) Open the **.tok** file. See [Opening an Appset file](#) for instructions. The Token view appears.



- 2) Double-click the Token value you wish to change (or select it, and click **Edit Value**).



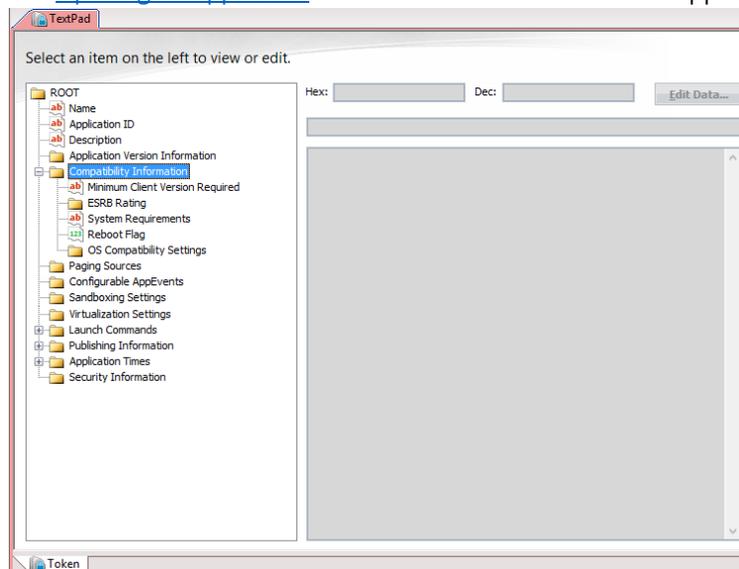
The **Enter the new value** dialog appears.

- 3) Enter the new value and click **OK**.

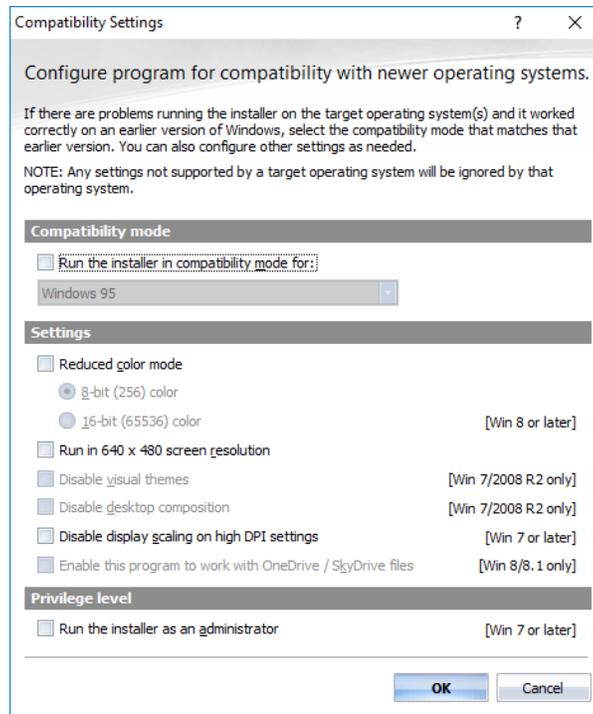
Modifying OS Compatibility Settings

To modify the OS compatibility settings of a Token file:

- 1) Open the **.tok** file. See [Opening an Appset file](#) for instructions. The Token view appears.



- 2) Expand **Compatibility Information**, and double-click **OS Compatibility Settings**.



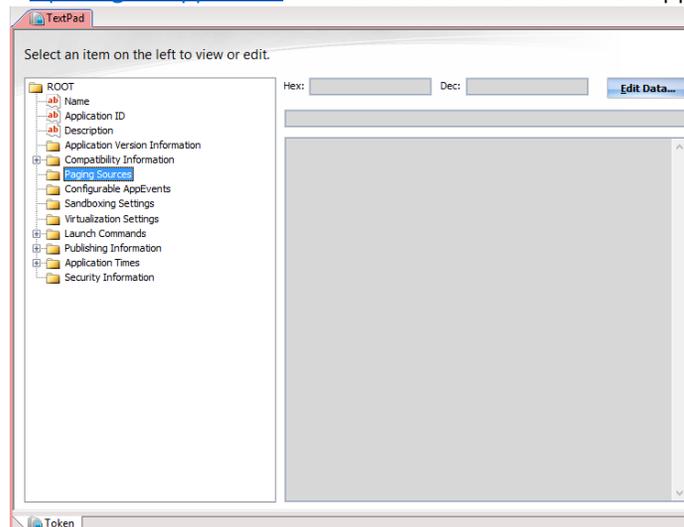
The **OS Compatibility Settings** dialog appears.

- 3) Select the desired setting for each set of operating systems and click **OK**.

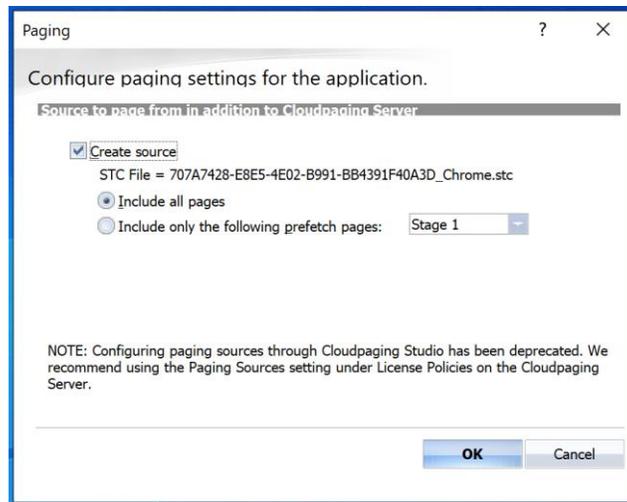
Modifying a Paging Source

To modify a Paging source settings of a Token file:

- 1) Open the **.tok** file. See [Opening an Appset file](#) for instructions. The Token view appears.



- 2) Double-click **Paging Sources**.



The **Paging** dialog appears.

- 3) Disable or enable sources as necessary and click **OK**.

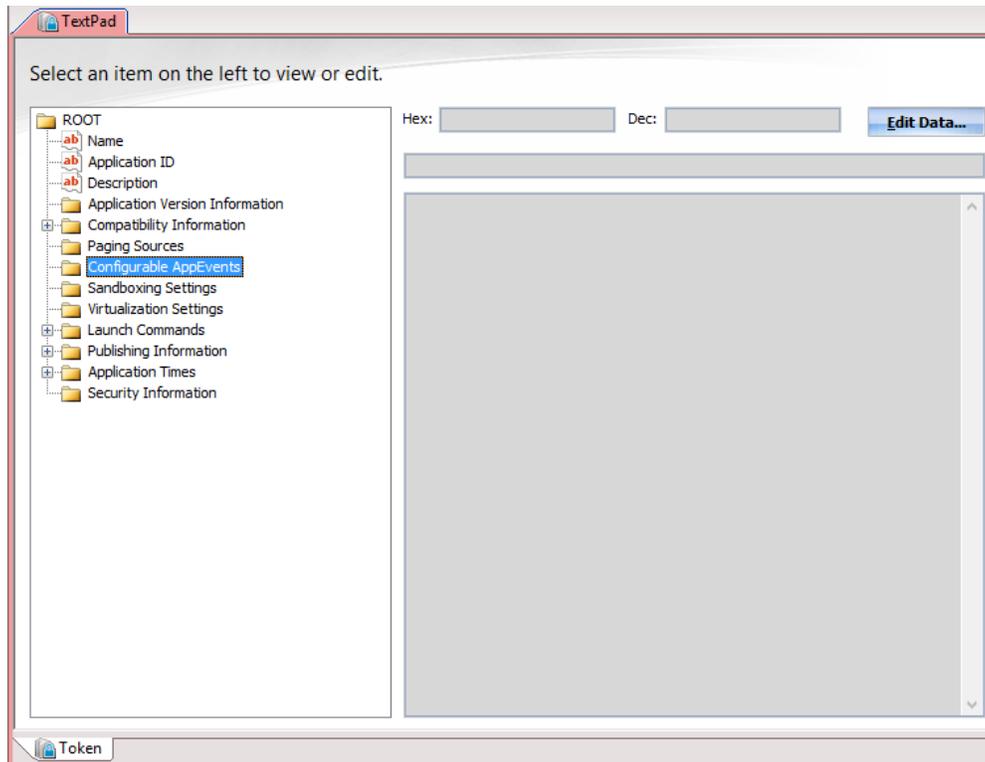
To create a new paging source:

- 1) Check Create source.
- 2) Then select one of two options:
 - **Include only the following prefetch page** – Indicates that the selected prefetch pages will be made available at the source location and add them to a new source STC file.
 - **Include all pages** – Indicates that all pages will be made available at the source location and add them to a new source STC file.
- 3) Enter a file name and path for the new STC file.

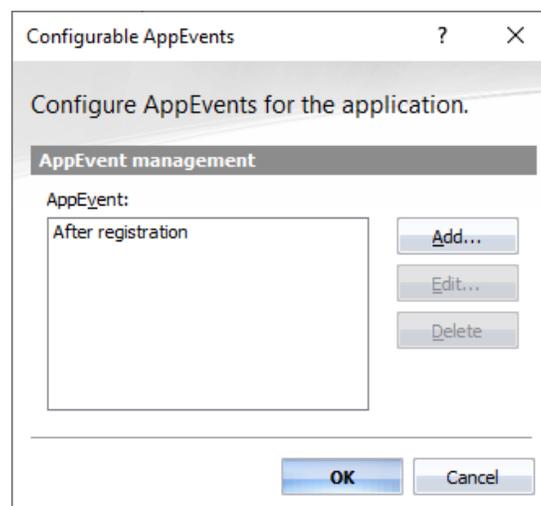
Modifying a Configurable AppEvent

To add a Configurable AppEvent to a token file:

- 1) Open the **.tok** file. See [Opening an Appset file](#) for instructions. The Token view appears.

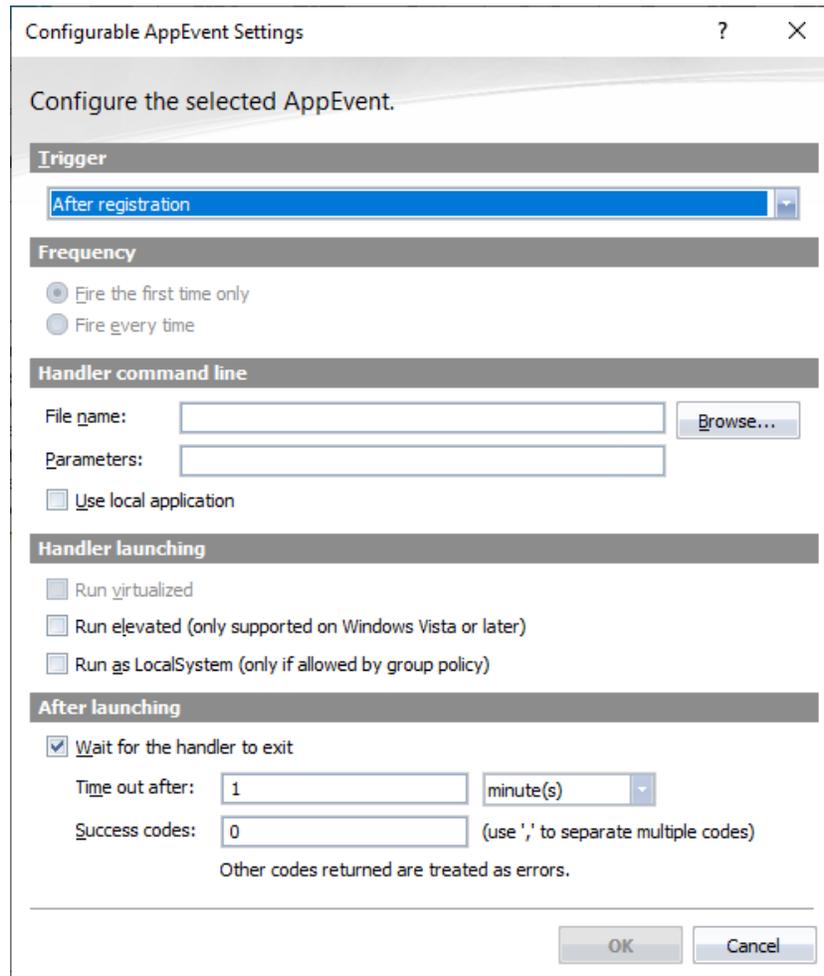


- 2) Double-click Configurable AppEvents.



The Configurable AppEvents dialog appears.

- 3) Click **Add**

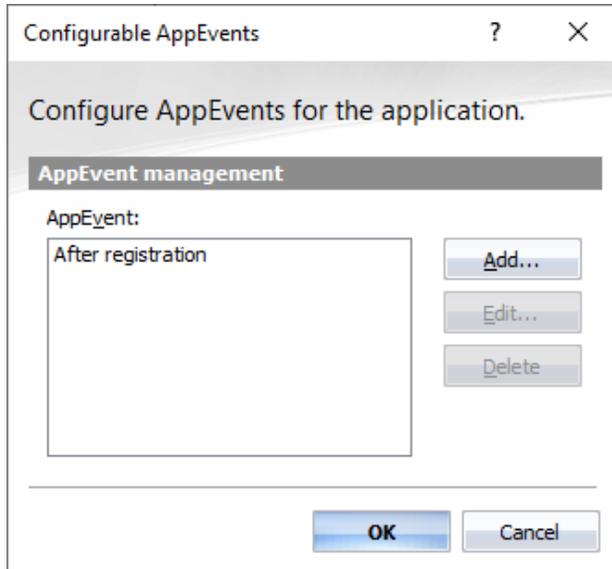


The Configurable AppEvents Settings dialog appears.

- 4) Enter the settings for the new Configurable AppEvent. For more information, see [To add an AppEvent.](#)

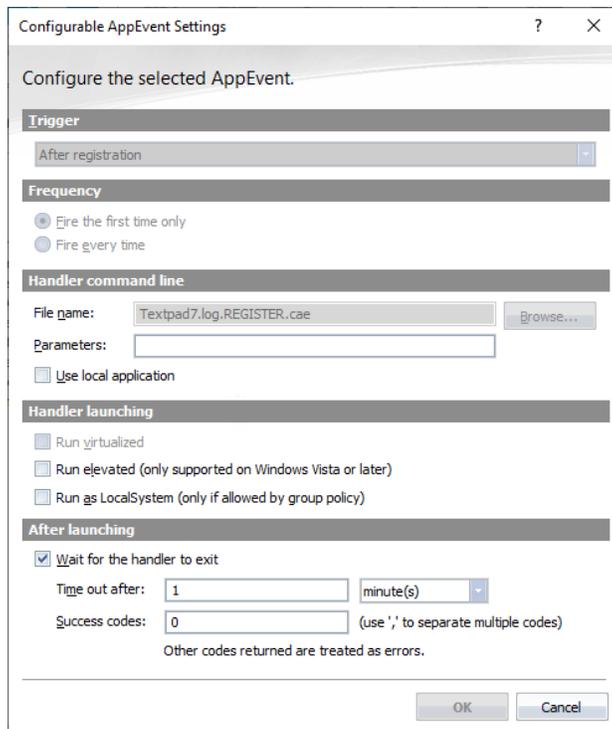
To Edit the Settings for an existing Configurable AppEvent

- 1) Open the **.tok** file. See [Opening an Appset file](#) for instructions. The Token view appears.
- 2) Double-click Configurable AppEvents.



The Configurable AppEvents dialog appears.

- 3) Select an existing AppEvent and click **Edit**



The Configurable AppEvents Settings dialog appears.

- 4) Edit the settings. For more information, see [To add an AppEvent](#).

NOTE

Cloudpaging Studio does not allow changing the handler of a CAE in the token workspace. If you need to change the handler, then you should remove the CAE and add it back with the new handler. See [To change the handler of a Configurable AppEvent:](#)

To remove a Configurable AppEvent from a token file:

- 1) Open the **.tok** file. See [Opening an Appset file](#) for instructions. The Token view appears.
- 2) Double-click **Configurable AppEvents**.
- 3) The **Configurable AppEvents** dialog appears.
- 4) Select an existing AppEvent and click **Delete**.

To change the handler of a Configurable AppEvent:

Cloudpaging Studio does not allow changing the handler of an existing CAE in the token workspace directly. If you need to change the handler of an existing CAE, you should first delete the CAE as described above then add it back as described in [To add a Configurable AppEvent to a token file:](#) and select a new handler.

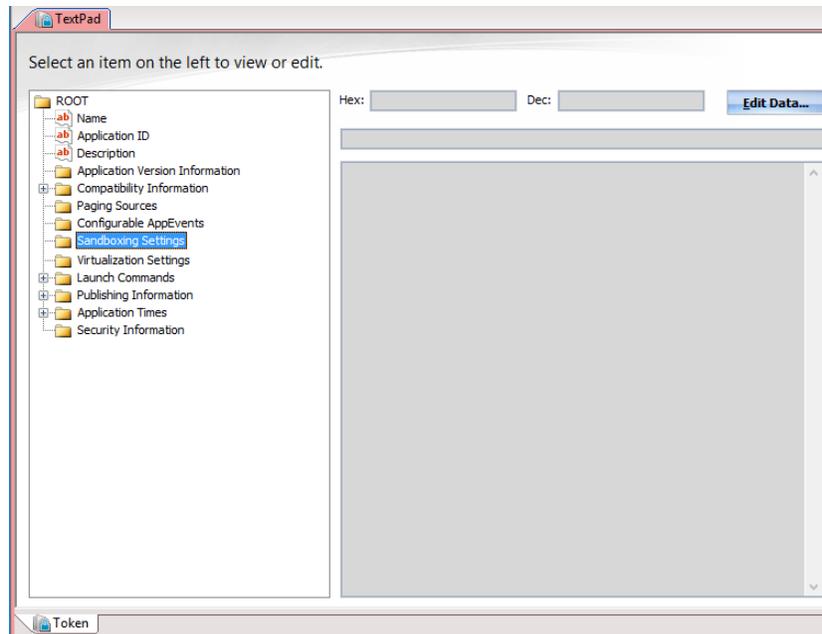
NOTE

Before removing the CAE, you may wish to write down or take a screenshot of all existing configuration settings so that you can set them again when you add back the CAE.

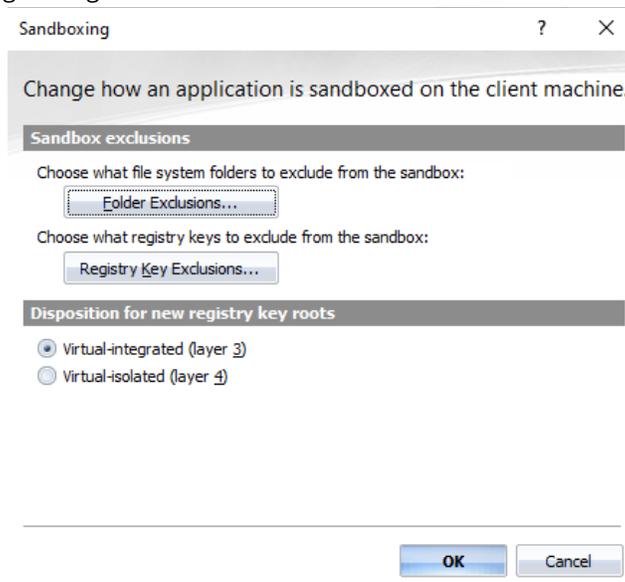
Modifying Sandboxing Settings

To modify the Sandboxing settings in a Token file:

- 1) Open the **.tok** file. See [Opening an Appset file](#) for instructions.
The Token view appears.



2) Double-click Sandboxing Settings.



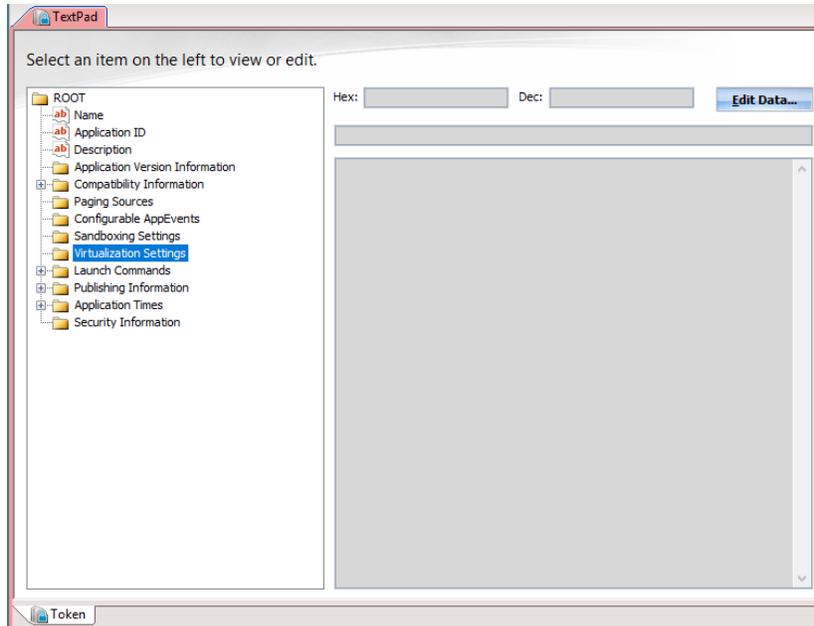
The Sandboxing Settings dialog appears.

For more information on using these settings, see [Sandboxing Settings](#).

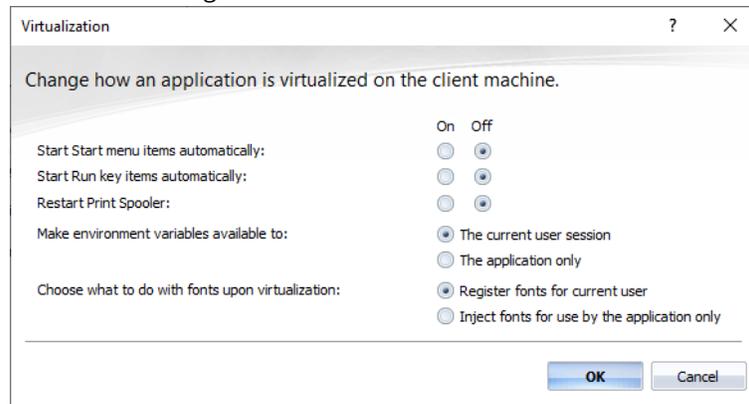
Modifying Virtualization Settings Flags

To modify the virtualization settings in a Token file:

1) Open the **.tok** file. See [Opening an Appset file](#) for instructions. The Token view appears.



- 2) Double-click the Virtualization Settings



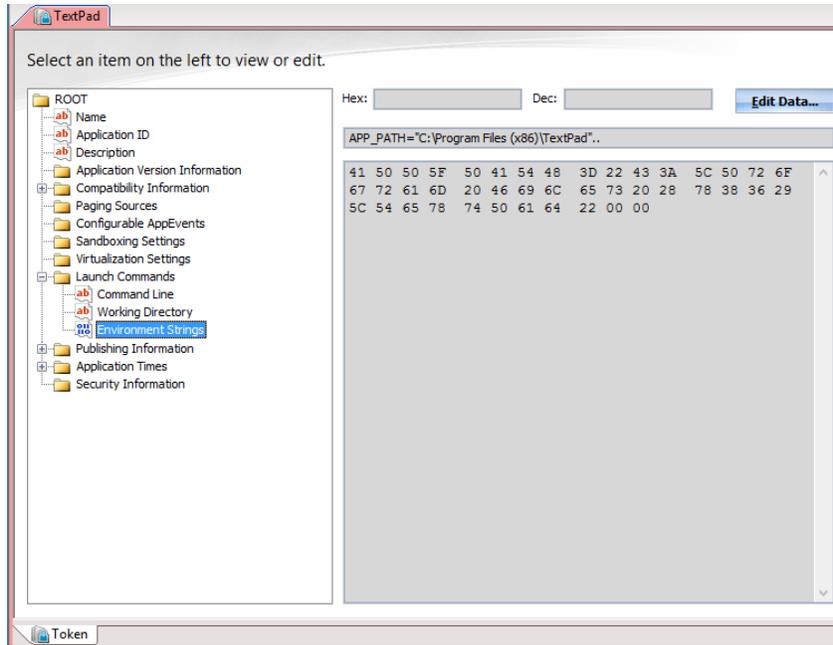
The Virtualization Settings dialog appears.

- 3) Change the desired settings. For more information on the setting on this dialog, see [Virtualization Settings](#).

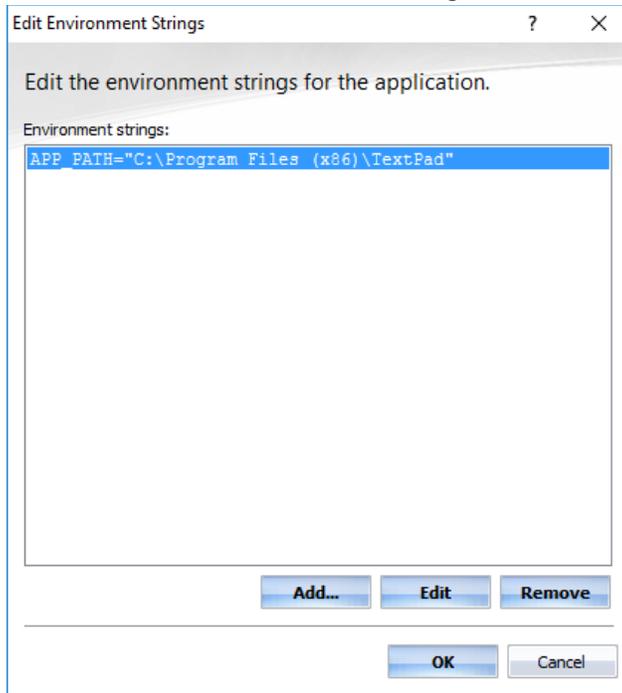
Modifying an Environment Setting

To modify an Environment setting of a Token file:

- 1) Open the **.tok** file. See [Opening an Appset file](#) for instructions. The Token view appears.

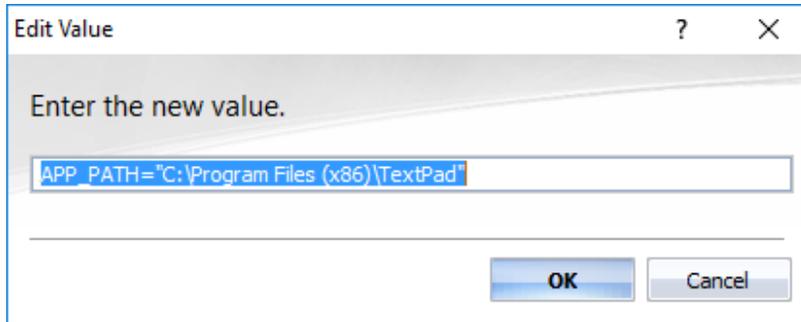


2) Expand Launch Commands, and double-click Environment Strings.



The Edit Environment Strings dialog appears.

Click Add or Edit as needed to display the Edit Value dialog.



- To add an Environment String, click **Add**.
- To edit an Environment String, click **Edit**.

The Enter the new value dialog appears.

Enter new values as needed, and click OK.

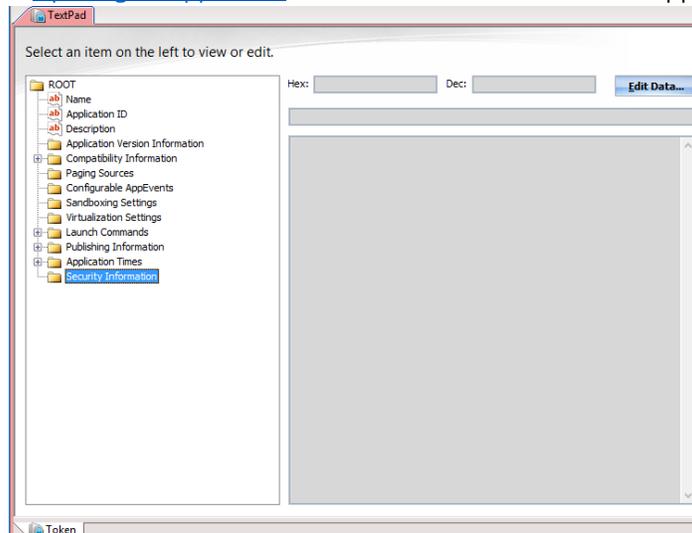
Deleting an Environment string:

- 1) Select the string from the **Edit Environment Strings** dialog.
- 2) Click **Remove**.

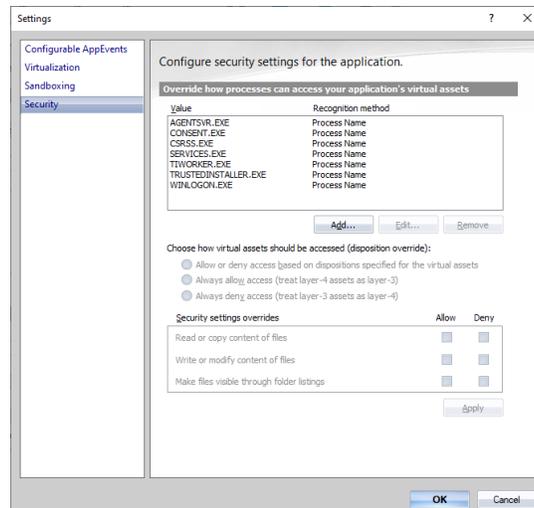
Modifying Security Override Settings

Modifying the security override settings of a Token file:

- 1) Open the **.tok** file. See [Opening an Appset file](#) for instructions. The Token view appears.



- 2) Double-click Security Information.



The Security Overrides dialog appears.

- 3) Choose the desired security overrides and click OK.

Saving an Appset File

After the AIB and/or TOK file has been updated, they need to be saved back into the appset (.stp) file format. You could also use this procedure to update the EULA HTML file or icon file.

To accomplish this requires a third-party compression utility called 7-zip, which can be found here: <https://www.7-zip.org/download.html>

Please be sure to have this utility installed. We recommend 7-Zip for its support of UTF-8 names and large files. It is a free open source utility.

Important

The built-in Windows file compression does not have this support and creating an appset this way could cause errors.

To save an appset file using the 7-Zip Command Line

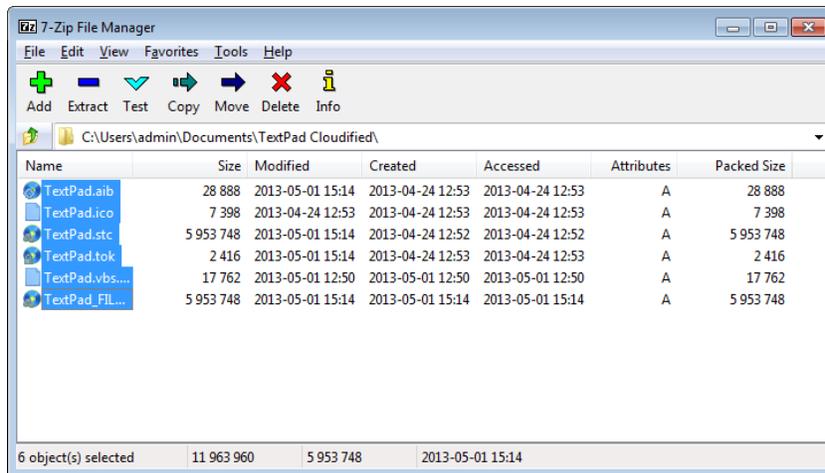
- 1) Open a command prompt window.
- 2) Navigate to the location of the STC, AIB, TOK, ICO, or EULA files.
- 3) Enter the following command line:

```
7z a -mcpu=on MyAppset.zip MyAppset.tok MyAppset.aib
MyAppset.stc MyAppset.ico
```

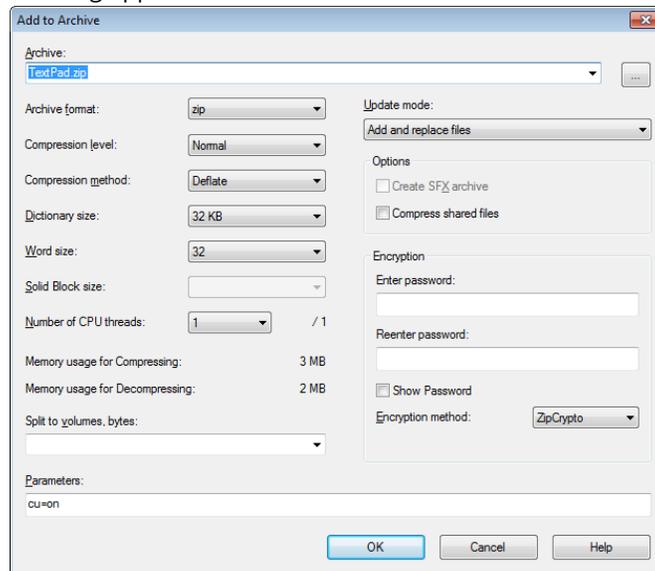
 Your appset may also include Configurable AppEvent files (.cae) files. These also needed to be included in the command line.
- 4) After 7-Zip creates the new ZIP archive, change the extension of the archive file to "stp."

To save the appset using the 7-Zip GUI

- 1) Start 7-Zip.
- 2) In the 7-Zip explorer window, navigate to the folder containing your appset files.
- 3) Select the appset files.



- 4) Click **Add**.
- 5) The 7-Zip Add to Archive dialog appears.



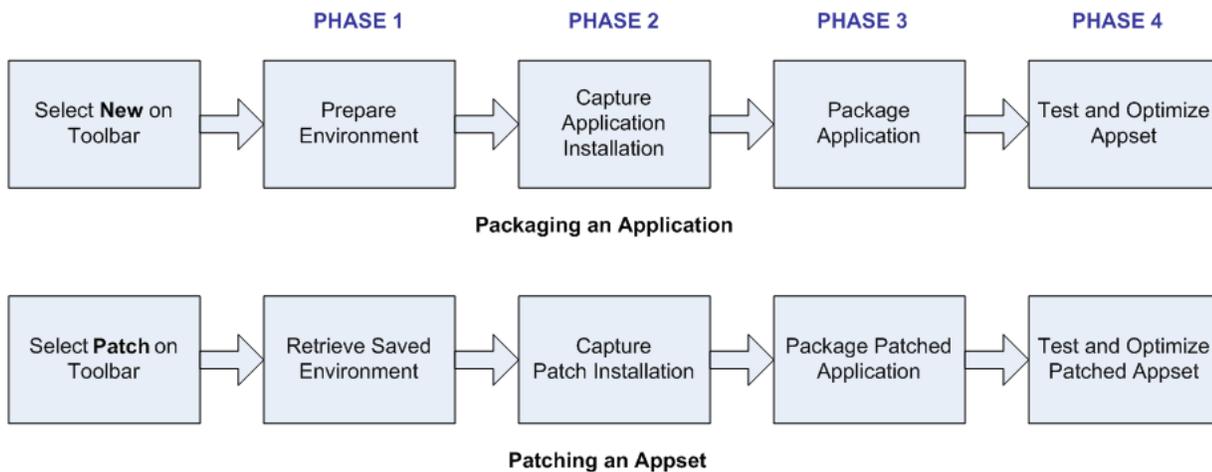
- 6) Enter "cu=on" into the Parameters text box.
- 7) Click **OK**.
- 8) After 7-Zip creates the new ZIP archive, change the extension of the archive file to "stp."

Important

You can also right click on the appset files from within Windows Explorer and use the context menu to launch 7-zip. In order for 7-zip to create the appset correctly, you must select **Add to Archive**. Do not use any of the other options which bypass the Add to Archive dialog above.

Chapter 9: Patching an Appset

Periodically, applications require program patches or release new versions. Cloudpaging fully supports both provisioning of new versions as major upgrades and/or patches to existing versions that have already been packaged. For new versions of software you should generate a new appset due to the potentially large number of differences between versions, and use the Major upgrade type to deploy the update. For patches, such as hotfixes or incremental updates, the existing appset should be updated to include the patch information. Patching an existing appset is the most efficient manner to update an existing application as only the delta changes between the previous version and new version are paged to the desktop. Patch information is captured by running the patch provided by the software vendor, and capturing the changes in the files and registry settings. An appset can contain multiple patch versions and upgrade types allowing for easy deployment or rollbacks for each individual patch version. See the *Cloudpaging Server Administration Guide* for details on upgrade types.



The following are explained in this chapter:

- Patching an Appset
- Modifying a Patched Appset
- Extracting an Application for Patching

Patching an Appset

Patching allows application updates to be captured and recorded in the appset. Appsets can contain multiple patches, which are added cumulatively. Each patch adds the required new files and registry settings to the STC file, and creates a new AIB file as well.

The following is required in order to create a patched appset:

- The environment that was used to create the current appset (copy of VM or image of physical PC)
- The **Cloudpaging Studio** project file (.stw) for the appset being patched.
- The appset file (.stp).

If you don't have access to the original environment, you can recreate the project file on a new machine by unpacking the appset file (.stp). For more information, see [Extracting an Application for Patching](#).

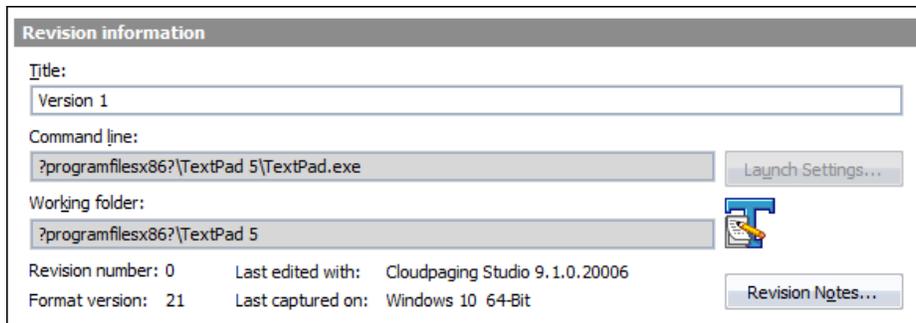
Important

*Make sure you have opened the correct project file before patching. And make sure that only one *.stw file is open, including the "Untitled*.stw" file.*

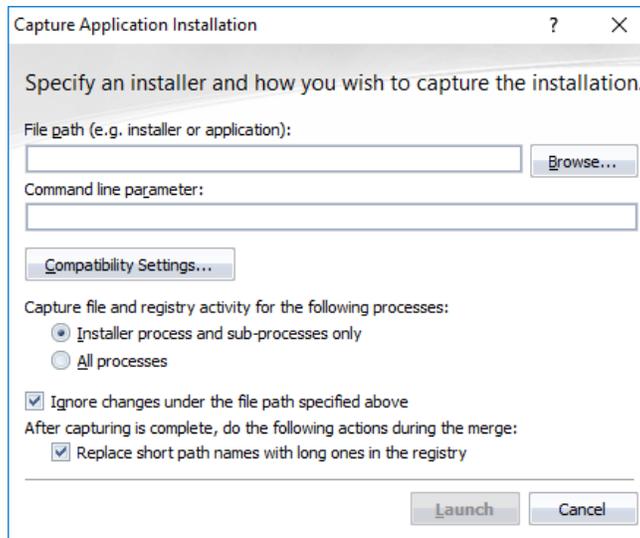
To patch an existing appset:

- 1) Either restore the base environment that was used to create the appset from a VM or a physical PC image, or extract the appset (.stp) file to a clean PC.
- 2) Launch **Cloudpaging Studio**.
- 3) Click **Patch** on the **Cloudpaging Studio** toolbar, navigate to the project file (.stw) for the appset, and click **Open**.
- 4) Enter a descriptive name in the Patch Title box. This could include the version and any other information that would enable users to be sure they had the latest version.

The launch settings button will be disabled, since you cannot change the command line for an existing application.



- 5) If desired, enter new revision notes for this patch. Click the **Revision Notes...** button to display the **Revision Notes** dialog. See [Revision Notes](#) for more information.
- 6) Click **Capture** on the **Cloudpaging Studio** toolbar. *Alternative:* Click **Application** on the menu bar, then select **Capture Application**. The **Capture Application Installation** dialog appears.



- 7) Click **Browse** and locate the software vendor's patch installer (.EXE) for the application.
- 8) Click **Open**, and then click **Launch**.
- 9) Select the destination folder, then follow the vendor's patch installation instructions to completion. For information about processes or compatibility settings, see [Start the Application Installation Capture Process](#).

The Files and Registry views will show which assets have been added, modified, or removed from the project by the new patch.

- Assets shown in **blue** are new. Installing the patch added them to project.
- Assets shown in **magenta** have been modified. Installing the patch overwrote the existing files with newer versions.
- Assets shown in **gray** were removed from the project.

- 10) Save the patched project file (.stw).
- 11) See [Cloudifying the Application](#) for cloudifying instructions.
- 12) See [Testing and Optimizing the Appset](#) for instructions on testing and optimizing the patched appset.

The patched appset is now ready to publish to **Cloudpaging Server**.

To add a prefetch to a patch appset:

When you create a patch, any prefetch files that have not changed from the previous version of the application will be carried over to the prefetch for the patch. Files that have changed will be removed.

Please follow the steps under [Chapter 5, Testing and Optimizing the Appset](#) for adding the new prefetch.

NOTE

Once you patch an appset, you can no longer modify the prefetch of the previous versions. You can only add or remove files from the prefetch of previous versions by editing the AIB files that correspond to those versions.

Modifying a Patched Appset

For instructions about modifying a patched appset, refer to [Chapter 7, Modifying Project Assets](#).

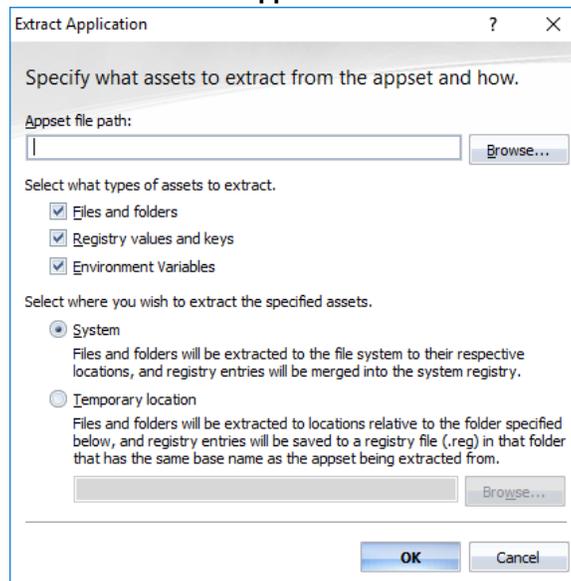
Extracting an Application for Patching

Cloudpaging Studio can extract assets of the original application from the appset, effectively recreating the original installation even if you don't have access to the original computer. Along with the original STW project file, you can then follow the steps above for patching an appset.

Extracting is also useful to examine or troubleshoot an appset. By extracting the appset to a new system, the application can be tested as a normal installation to ensure the appset contains the correct information.

To extract an application from an appset:

- 1) Start with a freshly installed Windows environment, running the same Windows version that was in use when the appset was originally created.
- 2) Copy the appset (.stp) file to a folder on this system.
- 3) Open **Cloudpaging Studio**.
- 4) Begin a new **Cloudpaging Studio** project.
- 5) Click the **Application** menu and select **Extract Application**



The **Extract Application** dialog appears.

- 6) Click **Browse** to locate the appset (.stp) file to be extracted.
- 7) Under **Extraction**, you can uncheck **File or Registry** if you do not need these parts of the appset extracted into the new project.

NOTE

In order to patch an appset, you must extract both file and registry assets when extracting. You can choose not to extract one or the other if you are examining or troubleshooting an appset.

- 8) Select the **Location** where **Cloudpaging Studio** will place the files for the extracted application.
 - **System** – Files are extracted to their original location (where they were stored on the PC when the appset was originally cloudified).
 - **Temporary** – Files are extracted to folder, which you specific by clicking the **Browse** button.

- 9) Click **OK** to begin extracting the application.

NOTE

Extraction does not run the Configurable AppEvents. After extraction, any commands added as a CAE can be launched manually, if necessary.

When **Cloudpaging Studio** is finished extracting the application, you can then open the original project file (STW), apply the patch, and re-cloudify the appset as described above.

Chapter 10: Troubleshooting

During the Testing and Optimizing phase, you may face certain complicating issues unique to the application which you are cloudifying. This chapter will describe the most common of these situations and their solutions, along with techniques for dealing with less common problems.

The following are explained in this chapter:

- Identifying Common Problems
- Solutions to Common Problems
 - Missing Shortcuts
 - Blank Shortcuts
 - File Associations Do Not Work
 - Application Fails to Launch
 - Printing Problems
 - Fonts Do Not Appear
 - Folder or Files Disappear from the Virtualized Application
 - Hide "Add/Remove Programs" Entry
 - Roaming and Folder Redirection Issues
 - Prerequisites
 - Unique User Accounts
 - Service Show as Running and Cannot Be Stopped
 - Windows Services or Drivers Do Not Start
 - DCOM and COM+ Services Not Remotely Accessible
 - Plug and Play Device Drivers
 - Alternative Data Streams
 - Anti-virus and Firewall Applications
 - Application Copy Protection
- Approach to Resolving Other Problems

Identifying Common Problems

Occasionally, there are errors that can be difficult to identify. If not all files or registry keys seem to have been captured, use the following troubleshooting steps (if applicable) to identify them:

- 1) Extract the appset file on a different PC.

This can be useful for a couple of reasons. It allows you to examine the file structure and registry keys to verify that everything necessary for the program to function is being captured. You can also extract the appset file on a clean PC to verify that the application will run on that computer. You can then verify that all the necessary files, registry keys, environment variables, fonts, and services are included in the appset. For more information, see [Extracting an Application for Patching](#).

- 2) Review the **Cloudpaging Player** Log file.

If the extracted appset worked correct, then most likely there is a problem with the settings. Viewing the **Cloudpaging Player** log can help to identify what the issues may be. Also check the following:

- Check that the disposition layer makes the file or key accessible to the application attempting to access it.
- Check that the security setting allows the file to be found and read.

- 3) Review the **Cloudpaging Studio** Log file.

This log will contain information on files and registry key captured and filtered from the appset. It will also contain information on any errors that may have been encountered. The **Cloudpaging Studio** can be viewed by clicking the **Application** menu and selecting **View Log**. Review the warning given during the capture about security descriptors, user accounts, etc. and will show a listing of files and registry keys not merged.

Solutions to Common Problems

While cloudifying an application will typically have successful results, there are times when the results may not match expectations. For example, did the application create a desktop shortcut and is the expectation that there should be a shortcut? Virtualization layer and security settings can also add to the complexity of how applications should behave for users. Below are solutions to some of the most common problems.

Missing Shortcuts

Most applications create shortcut files (.lnk) on the Desktop or in the Start menu. These files can be located in the **Cloudpaging Studio** File view located in the **CommonPrograms** or **UserProgram** folder for the **Start** menu and located in the **CommonDesktop** or **UserDesktop** folder for the Windows desktop. You may add or remove shortcut files as desired to these folders.

If the shortcut files for the cloudified application do not appear after virtualizing the appset, then do the following:

- 1) In order for the shortcut to appear on the **Start** menu, verify that the shortcut file is located in the **CommonPrograms** or **UserPrograms** folder. In order for the shortcut to appear on the Windows desktop, verify that the shortcut file is located in the **CommonDesktop** or **UserDesktop** folder.
- 2) Be sure that the Disposition layer for the shortcut files and folders are set to **Integrated** (Disposition layer 3). This will allow shortcut to be accessible on the OS of the Target PC.
- 3) Check that the security setting on the shortcut files do not have "**Hide from Folder Listing**" selected. If this option is selected, the shortcut files will not be visible on the Target PC.

Blank Shortcuts

If the Desktop icon for the application shows as a blank default Windows icon, the executable file for the application is probably set to "Prevent Copying" in the AIB file. This is normal behavior with this setting and the application will function normally.

File Associations Do Not Work

Windows manages file extension associations to applications under the HKEY_CLASSES_ROOT hive of the Windows registry. For file associations to work the file type must be set in the registry and the application executable must be visible on the Target PC. To fix file association issues, do the following:

- 1) Ensure that the file extension exists in the **Cloudpaging Studio** Registry view. If the extension is missing, then it might not have been setup by the application. Either manually add the extensions or use re-capture the application with "All processes" selected, run the application, and configure the default associations.
- 2) Be sure that all the extensions are set to **Integrated** (Disposition layer 3) so that they are accessible on the Target system.
- 3) Change the application main executables to **Integrated** (Disposition layer 3) so that the associated file can be opened by the application.

Application Fails to Launch

Sometimes an application can fail to launch when the command line is incorrect; the command line may have to be specified with a different executable or may need additional launch parameters. Check the original application shortcut, typically found in the Start menu, and use the same command line and parameters in Studio.

Another problem may be that some applications modify the local machine's PATH environment variable during installation or after launches for the first time. There may be some system specific entries included in the PATH variable upon cloudifying. Ensure that it contains no system specific entries. Target systems will have many different configurations and it is important to leave only the entries that are specific to the application being cloudified.

For example: **PATH = %PATH%;C:\ApplicationPath1;C:\ApplicationPath2**

To check if an application populates its own PATH environment variable after capturing an installation, go to the Environment view to check if the environment variable named "PATH" exists.

Printing Problems

Some application will install new printer devices, which can be included in an appset. In the Files view there is a folder named **spool** which is a subdirectory found under the SystemX86Dir system folder. Applications that come with printer drivers may not work properly if this folder is removed from the Files view. Within the Registry view, the registry key **HKEY_LOCAL_MACHINE \ SYSTEM \ CurrentControlSet \ Control \ Print \ Environments \ Windows NT x86 \ Drivers** contains references to the printer drivers in the spool folder. If a new printer does not appear, be sure the **Restart Print Spooler** option under the **Virtualization** screen of the **Settings** dialog has been selected.

Fonts Do Not Appear

Most applications come with their own program fonts. If an application is cloudified with fonts set to **Injected** under the **Virtualization** settings, then running the paged application from a shortcut will cause the fonts to not

appear. Private fonts are only injected into the paged application if it is launched from **Cloudpaging Player**. To make fonts available to all applications, change the **Font** setting to **Register**. In addition to the Fonts view, registered fonts can be found in the following locations:

- In the Files view, the fonts directory can be found at **SystemFonts**
- In the Registry view, the fonts can be found at **HKEY_LOCAL_MACHINE \ SOFTWARE \ Microsoft \ Windows NT \ CurrentVersion \ Fonts**

Folder or Files Disappear from the Virtualized Application

If you happen to cloudify an application to an **install folder** that has the same name as a folder on the local PC, then the local folder may disappear. This can also affect files under common folders marked with disposition layer 3 for integrated, but not merged (see [Merged and Non-merged Folders](#)). To solve this issue, be sure to install the application to a unique install folder name. You may also need to mark files in common folders as isolated if desired. To fix the existing appset:

- 1) Rename the **Install folder** to be more unique.
- 2) Set specific files in common locations to be **layer 4** - isolated or merged.

Hide "Add/Remove Programs" Entry

Many applications will place an entry in the "Add or Remove Programs" listing of the control panel (or "Programs and Features" on Windows 7). This can be confusing as the application is virtual and cannot be uninstalled. If you wish to hide this entry, take the following steps:

- 1) Under the Registry view, locate the following key **HKEY_LOCAL_MACHINE \ SOFTWARE \ Microsoft \ Windows \ CurrentVersion \ Uninstall**
- 2) Remove it from the workspace.

Roaming and Folder Redirection Issues

Some Enterprise deployments may use "Roaming Profiles" and/or "Folder Redirection" to preserve user application settings. For Roaming Profile support, it is important to setup UEM support on the **Cloudpaging Player**. **Cloudpaging Player** will then allow HKEY_CURRENT_USER registry settings and the %appdata% folder contents for paged applications to roam properly. Any limitations with Windows Roaming Profiles will still apply.

With Folder Redirection to a network share (via Mapped drive or a UNC path), only Layer-1 (recommended) and/or Layer-2 files are allowed. If any Layer-3 or Layer-4 files are located in these folders, then the appset will fail to activate. For example, if the %appdata% folder is being redirected to [\\fileshare\user1\appdata\](#), then the appset template folder ?roamingappdata? must only contain Layer-1 files and folders. Redirection to another local disk supports virtual files (Layer-3 or Layer-4).

Important

With folder redirection, the network folder must exist prior to activating the application. For example, if %appdata% is mapped to n:\{username}\appdata, then the "appdata" folder must already exist on the server.

Prerequisites

Many applications require the presence of specific runtime library or a minimum version of the .NET Framework.

Unless you are certain that the prerequisites will be present on the Host PCs, they should be installed and captured prior to capturing the main application. If possible, set the default disposition for new assets to level 4 before capturing. (see [Globally Changing Asset Dispositions](#) for more information).

It may be necessary to change the security override settings for some processes to keep the assets for a dependency sufficiently isolated. Also, the disposition for any of the following captured assets should be set to layer-4 as previously mentioned:

```
<WindowsDir>\winsxs\ all folders and files  
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\SideBySide\  
HKEY_LOCAL_MACHINE\Software\Classes\Installer\Win32Assemblies\
```

If an application requires the presence of a specific service pack version of the operating system, you should create a Configurable AppEvent to check for its presence on the Target PC after the application is registered and alert the user if it is not installed. See [About AppEvent Scripts and Executables](#) for more information.

Unique User Accounts

Some applications create a special user account when you install them. **Cloudpaging Studio** cannot capture user accounts, because these are not transferable between computers. You need to create an AppEvent to programmatically replicate the user accounts and password on the host computer when **Cloudpaging Player** virtualizes the application.

NOTE

User Accounts may appear as a security identifier (SID) in the registry. For example, S-1-5-18 is the local system account. Please see <http://support.microsoft.com/kb/243330> for more details.

Service Show as Running and Cannot Be Stopped

All executable files that are virtual (layer-3 or layer-4) will be tracked for usage and shown with a “Running” status when being used. This applies to services as well. To not be tracked for usage, the files (i.e. Service ImagePath) would need to be mark for a layer-2 disposition.

In addition, some services may be configured to automatically restart if interrupted. This can be found by running the **services.msc** command, selecting **properties** on the service in question, and going to the **Recovery** view. If the **Subsequent failures** setting is not set to **Take No Action**, then **Cloudpaging Player** will not be able to stop the service and remove the application. To correct this it is best to remove the registry value called **FailureActions** under HKEY_LOCAL_MACHINE \ SYSTEM \ CurrentControlSet \ Services in the Register view for that service.

Windows Services or Drivers Do Not Start

In most cases, **Cloudpaging Player** handles initialization and starting services and drivers without further intervention. If a service needs to start using a unique user account (this normally defaults to LocalSystem), then the appset will need an AppEvent to handle creating this account.

Important

Please be aware that on some systems services have security descriptors that are enforced more strictly. When adding a new service, it is important to match the security descriptor of the installed version. Use **sc sdshow <service_name>** in command prompt to display security descriptor set for the service. Marking an appset to **require system reboot** after adding a service will typically correct the security descriptors.

If the application has boot or system drivers, then mark all the files and registry keys for those services and/or drivers as Layer 2 (Installed-temporary disposition).

- 1) Under the Services view, right-click the service to select **Service Key Disposition** and mark it as **Layer 2**.
- 2) Right-click the same service to select **Image File Properties** and mark it as disposition **Layer 2**.

Important

Marking files for drivers and/or services as layer 2 may causes problems if the appropriate registry keys are also not set as layer Be sure to perform both steps.

- 3) Exclude all LEGACY registry keys for that service/driver and any specific Enum keys:
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Enum\Root
HEKY_LOCAL_MACHINE\ System\CurrentControlSet\Services\{NAME}\Enum
- 4) In the Advance settings, select **Require system reboot**. If you wish to avoid a reboot, then a CAE may be used to start the driver directly using the command:
sc.exe start <name of driver>

For more information on how services work, please refer to the Microsoft Knowledge Base:

<http://support.microsoft.com/kb/251192>

NOTE

Typically a system driver cannot be stopped and removed from a system without a reboot. Cloudpaging Player will properly remove these files from the system but the driver will still be loaded until a reboot occurs.

DCOM and COM+ Services Not Remotely Accessible

DCOM and COM+ are technologies that extend COM to allow remote processes to use local COM services. They are managed by system-level host containers, which means their assets (files and keys) must be layer 1, 2 or 3, just like COM. Security overrides must be set for these host containers to allow remote access (e.g., from another Windows session) to DCOM and COM+ services.

The host container for DCOM is SVCHOST.EXE, which is already set with the necessary security overrides by default. So, if your application contains DCOM services, these should work as is with the default packaging security settings. If they do not, then the SVCHOST.EXE process may have been removed from the process list in the **Security** screen of the **Settings** dialog. In this case it should be reinstated with the appropriate settings, as described below.

The host container for COM+, on the other hand, is DLLHOST.EXE, which is not set with any security overrides by default. So, if your application contains any COM+ services, you will need to manually add DLLHOST.EXE to the process list in the **Security** screen of the **Settings** dialog (see [Security Settings](#)) and configure it identically to SVCHOST.EXE, as follows:

- 1) Set the process to be recognized by Process Name.
- 2) Select the Always allow access (treat layer-4 assets as layer-3) disposition override.
- 3) Check Allow for the Read or copy content of files security setting.

Plug and Play Device Drivers

With all recent versions of Windows, Microsoft introduced what is called the Driver Store, a protected repository for trusted device drivers. These platforms will require a Configurable AppEvent to setup the device driver.

64-bit versions will be different from 32-bit device drivers and also may require additional configuration.

Important

Please do not attempt to set files in the Driver Store to layer 1 or This will cause virtualization to fail.

The utility **pnputil.exe** runs from a command line and its primary purpose is to add drivers to the Driver Store and remove them. The basic syntax for adding drivers using the utility is:

```
pnputil.exe -i -a "C:\<folder>\driver.inf"
```

The best approach to cloudifying a driver for Windows 7 or later:

- 1) Capture the application and drivers as normal. Exclude the folder <WindowsDir>\inf\.
- 2) Exclude all virtual files used by the device driver. To know which files are use, first run the command devmgmt.msc that will open the Device Manager window. Find the device driver being captured and double-click it. On the **Driver** tab, click the **Driver Details** button and a dialog will show a list of files that should be excluded.
- 3) For TWAIN devices, mark all files located under <WindowsDir>\SSDriver\ as layer-2.
- 4) Create a Configurable AppEvent, calling **pnputil.exe** using the syntax shown above. The CAE should use the following settings:
 - The trigger should be set to **After virtualization**.
 - The path should be set to **%SystemRoot%\system32\pnputil.exe** and mark for **Use local application**.
 - Set the handler launching to **Run elevated**.
- 5) The driver to register will be normally found under c:\windows\system32\DriverStore\FileRepository as an INF file. Register all the INF files.
- 6) You will need to alert the user that, when the CAE runs the **pnputil.exe**, Windows will display a security warning. They will need to click the button to allow the installation.

NOTE

Device drivers will remain on the system even after the application has been removed by Cloudpaging Player. This is expected with device drivers.

Alternative Data Streams

Users may receive warning messages about Alternate Data Streams (ADS) when **Cloudpaging Player** virtualizes an application. In most cases, these messages are not a serious problem and may be ignored. In rare cases, the application may require the existence of an ADS for a particular asset. In this case, the appset should exclude the asset from the sandbox. Set the disposition of the asset to level 2 or lower.

Anti-virus and Firewall Applications

While it is possible to capture and create appsets for various anti-virus and firewall applications, please be aware that these applications include drivers and may limit how **Cloudpaging Player** works. For example, if the application is a Firewall, it may attempt to block **Cloudpaging Player** from properly fetching data over the internet by default. Be sure to add **Cloudpaging Player** components to any exceptions rules (e.g., Add streamingcore.exe to a firewall allow list) when the appset is created.

Application Copy Protection

Many applications employ some form of copy protection to prevent theft and enforce proper licensing. Cloudpaging does not attempt to remove such protection. It is ideal to cloudify an application using a serial number that is based for high volume licensing (e.g., Microsoft products use Volume License Keys) so that multiple systems can activate with one key. Otherwise, the application will prompt for licensing as it would normally and/or it may not be possible for every system to activate the application and to make it fully functional.

It is best to test the appset on other systems to ensure there is no form of license enforcement that you are not already aware of. If the appset does not work or prompts for licensing, then retest on an image, or snapshot, of the same system used to cloudify. If the application binds to hardware, then it should work on the same system. Please contact the software ISV for licensing models that they support.

Approach to Resolving Other Problems

If you have identified a problem that you could not solve with the above procedures, use the following guidelines to resolve the problem:

- Cloudify the application again with the “All processes” option. These are typically due to files or registry keys that are either not captured by **Cloudpaging Studio** or have been removed from the appset. If you suspect that not all files or registry keys were captured, it is best to capture the application again and specify the “All processes” option. If the problem persists, then additional troubleshooting will be required.
- Cloudify the application again without converting short path names to long names. A few applications still require file references by the old DOS-style file and folder names. If these files are converted during the merge process, it could break the cloudified application. Capture the application, but uncheck **Replace short path names with long ones in the registry**.
- Open the Event Viewer and review the Application Logs. Run Event Viewer (**Windows Control Panel > Administrative Tools**) on the Target PC and view the Application Logs and look a detailed description of Errors or Warnings. Many times, the logs will provide information about a file or registry key that cannot be located or accessed.
- Use Process Monitor to view files, registry, and process information. Process Monitor, a utility that is available from Microsoft’s website, monitors files, registry, and process information of the application. Process Monitor will display a large amount of system information, so start by excluding all processes other than the paged application. Next, launch the

paged application and search for “NOT FOUND” results. These will indicate that the application was not able to locate a file or registry key.

- Dump the file and registry lists from the AIB file editor.

Compare these text files to similar outputs from known good appsets for the same or a similar application. Any differences may help to highlight the problem. See [Modifying File Settings](#) and [Modifying Registry Settings](#) for more information.

Appendix A: Folder Template Mapping

Folder Mappings

Folder Name	Template	Default Path
AppData	?roamingappdata?	C:\Users\ <username>\AppData\Roaming\</username>
CommonAdminTools	?commonadmintools?	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Administrative Tools
CommonDesktop	?publicdesktop?	C:\Users\Public\Desktop
CommonDocuments	?publicdocuments?	C:\Users\Public\Documents
CommonDownloads	?publicdownloads?	C:\Users\Public\Downloads
CommonGamesTasks	?publicgametasks?	C:\ProgramData\Microsoft\Windows\GameExplorer
CommonMusic	?publicmusic?	C:\Users\Public\Music
CommonOEMLinks	?commonoemlinks?	C:\ProgramData\OEM Links
CommonPictures	?publicpictures?	C:\Users\Public\Pictures
CommonProgramFiles X64	?programfilescommonx64?	See remark
CommonProgramFiles X86	?programfilescommonx86?	See remark
CommonPrograms	?commonprograms?	C:\ProgramData\Microsoft\Windows\StartMenu\Programs
CommonStartMenu	?commonstartmenu?	C:\ProgramData\Microsoft\Windows\StartMenu\Programs
CommonStartup	?commonstartup?	C:\ProgramData\Microsoft\Windows\StartMenu\Programs\StartUp
CommonTemplates	?commontemplates?	C:\ProgramData\Templates
CommonVideos	?publicvideos?	C:\Users\Public\Videos
DeviceMetaDataStore	?w7_devicemetadastore?	C:\ProgramData\Microsoft\Windows\DeviceMetadataStore
DocumentsLibrary	?w7_documentslibrary?	C:\Users\ <username>\AppData\Roaming\Microsoft\Windows\Libraries\Documents.library-ms</username>
ImplicitAppShortcuts	?w7_implicitappshortcuts?	C:\Users\ <username>\AppData\Roaming\Microsoft\Internet</username>

		Explorer\Quick Launch\User Pinned\ImplicitAppShortcuts
Libraries	?w7_libraries?	C:\Users\ <username>\AppData\Roaming\Microsoft\Windows\Libraries</username>
LocalAppData	?localappdata?	C:\Users\ <username>\AppData\Local\</username>
LocalizedResourcesDir	?localizedresourcesdir?	C:\Windows\resources\ <code page><="" td=""> </code>
PictureLibrary	?w7_pictureslibrary?	C:\Users\ <username>\AppData\Roaming\Microsoft\Windows\Libraries\Pictures.library-ms</username>
ProgramData	?programdata?	C:\ProgramData
ProgramFilesX64	?programfilesx64?	See remark
ProgramFilesX86	?programfilesx86?	See remark
PublicDir	?public?	C:\Users\Public
PublicLibraries	?w7_publiclibraries?	C:\Users\Public\Libraries
PublicRingtones	?w7_publicringtones?	C:\ProgramData\Microsoft\Windows\Ringtones
RecordedTVLibrary	?w7_recordedtvlibrary?	C:\Users\Public\Libraries\RecordedTV.library-ms
ResourceDir	?resourcedir?	C:\Windows\Resources
Ringtones	?w7_ringtones?	C:\Users\ <username>\AppData\Local\Microsoft\Windows\Ringtones</username>
SampleMusic	?samplemusic?	C:\Users\Public\Music\Sample Music
SamplePictures	?samplepictures?	C:\Users\Public\Pictures\Sample Pictures
SamplePlaylists	?sampleplaylists?	N/A
SampleVideos	?samplevideos?	C:\Users\Public\Videos\Sample Videos
SideBarDefaultParts	?sidebardefaultparts?	N/A
SystemDir	?system?	C:\Windows\System32
SystemFonts	?fonts?	C:\Windows\Fonts
SystemX86Dir	?systemx86?	See remark
UserAdminTools	?admintools?	C:\Users\ <username>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Administrative Tools</username>
UserCDBurning	?cdburning?	C:\Users\ <username>\AppData\Local\Microsoft\Windows\Burn\Burn</username>
UserContacts	?contacts?	C:\Users\ <username>\Contacts</username>
UserCookies	?cookies?	C:\Users\ <username>\AppData\Roaming\Microsoft\Windows\Cookies</username>
UserDesktop	?desktop?	C:\Users\ <username>\Desktop</username>

UserDocuments	?documents?	C:\Users\ <username>\ Documents</username>
UserDownloads	?downloads?	C:\Users\ <username>\Downloads</username>
UserFavorites	?favorites?	C:\Users\ <username>\Favorites</username>
UserGameTasks	?gametasks?	C:\Users\ <username>\AppData\Local \Microsoft\Windows\GameExplorer</username>
UserInternetCache	?internetcache?	C:\Users\ <username>\AppData\Local\ \Microsoft\Windows\Temporary Internet Files</username>
UserLinksDir	?links?	C:\Users\ <username>\Links</username>
UserLocalAppDataLow	?localappdatalow?	C:\Users\ <username>\AppData\LocalLow\</username>
UserMusic	?music?	C:\Users\ <username>\Music</username>
UserNetworkShortcuts	?nethood?	C:\Users\ <username>\AppData\Roaming\Microsoft\Windows \Network Shortcuts</username>
UserOriginalImages	?originalimages?	C:\Users\ <username>\AppData\Local\Microsoft\Windows Photo Gallery\Original Images</username>
UserPhotoAlbums	?photoalbums?	C:\Users\ <username>\Pictures\Slide Shows</username>
UserPictures	?pictures?	C:\Users\ <username>\Pictures</username>
UserPinned	?w7_userpinned?	C:\Users\ <username>\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch\User Pinned</username>
UserPrinterShortcuts	?printhood?	C:\Users\ <username>\AppData\Roaming\Microsoft\Windows \Printer Shortcuts</username>
UserProfile	?profile?	C:\Users\ <username>< td=""> </username><>
UserProfiles	?userprofiles?	C:\Users
UserPrograms	?programs?	C:\Users\ <username>\AppData\Roaming\Microsoft\Windows \Start Menu\Programs</username>
UserProgramFilesCommon	?w7_userprogramfiles?	C:\Users\ <username>\AppData\Local\Programs</username>
UserPrograms	?w7_userprogramfile scommon?	C:\Users\ <username>\AppData\Local\Programs\Common</username>
UserQuickLaunch	?quicklaunch?	C:\Users\ <username>\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch</username>
UserRecent	?recent?	C:\Users\ <username>\appdata\roaming\microsoft\window </username>\appdata\roaming\microsoft\window s\Recen
UserSavedGames	?savedgames?	C:\Users\ <username>\Saved Games</username>
UserSearches	?savedsearches?	C:\Users\ <username>\Saved Searches</username>
UserSendTo	?sendto?	C:\Users\ <username>\AppData\Roaming</username>

		\\Microsoft\Windows\SendTo
UserStartMenu	?startmenu?	C:\Users\ <username>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs</username>
UserStartup	?startup?	C:\Users\ <username>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup</username>
UserTemplates	?templates?	C:\Users\ <username>\AppData\Roaming\Microsoft\Windows\Templates</username>
UserVideos	?videos?	C:\Users\ <username>\Videos</username>
VideosLibrary	?w7_videoslibrary?	C:\Users\ <username>\AppData\Roaming\Microsoft\Windows\Libraries\Videos.library-ms</username>
WindowsDir	?windows?	C:\Windows

* N/A means not supported on the platform

Directory Template	Operating System	Default Path
?programfilesx86?	32 bit	C:\Program Files
	64 bit	C:\Program Files (x86)
?programfilesx64?	32 bit	N/A*
	64 bit	C:\Program Files
?programfilescommonx86?	32 bit	C:\Program Files\CommonFiles
	64 bit	C:\Program Files(x86)\Common Files
?programfilescommonx64?	32 bit	N/A*
	64 bit	C:\Program Files\Common Files
?system?	32 bit	C:\Windows\System32
	64 bit	C:\Windows\System32
?systemx86?	32 bit	C:\Windows\System32
	64 bit	C:\Windows\Syswow64

* N/A means not supported on the platform

Appendix B: Editing Configuration Files

The files detailed in this appendix control some of the default behaviors of **Cloudpaging Studio** or default settings for the cloudified application. Editing these files allow you to control which elements of an application are captured and how the final appset interacts with the host PC where **Cloudpaging Player** is installed. These files are installed into the \lib sub-folder of the main **Cloudpaging Studio**, so the standard path to these files is:

```
[Install Path]\lib
```

There are eleven .Dat files in this folder. These are all text files, which can be edited with Windows Notepad or another text editor.

Conventions

Comment lines begin with a "#" character. Any non-blank line that doesn't begin with this character is considered a separate, active item and **Cloudpaging Studio** will attempt to act up on the information contained in that line.

Each file contains default entries designed to work in most situations. These also function as examples for adding new entries.

preloadrehives.dat

This file defines registry hives that must be loaded prior to capturing. Each entry specifies a hive to be loaded into the system registry using a RegLoadKey call.

Hives that are normally loaded by system at boot time (e.g. HKEY_LOCAL_MACHINE\Software) need not be listed in this file.

Example

```
HKEY_LOCAL_MACHINE\COMPONENTS %windir%\System32\Config\Components
```

procexcluded.dat

This file defines processes that will be ignored during Capture, even when the capture is launched with the "Capture All Processes" option checked.

The structure of each entry consists of the absolute path to the process executable to be excluded.

```
C:\WINDOWS\SYSTEM32\SEARCHINDEXER.EXE
```

filefilt.dat

This file defines files and folders that will be filtered out of the project file during the capture and merge process.

Each entry in this file consists of NAME followed by the path.

Example:

```
USERDATA_1 %userprofile%\NTuser.dat
```

NOTE

This file is loaded once every time a new project is created. Subsequently, loaded information from this file will be saved with the new project. Reopening an existing project will not reload this file and information will be reloaded from the saved project. In other words, changes to this file will affect only future projects when they are created.

regfilt.dat

This defines registry paths to which changes are excluded from the capture and merge process.

Each entry consists of a filter name (FILTER_001, FILTER_002), followed by white space and the full registry path of the key to be filtered.

NOTE

This file is loaded once every time a new project is created. Subsequently, loaded information from this file will be saved with the new project. Reopening an existing project will not reload this file and information will be reloaded from the saved project. In other words, changes to this file will affect only future projects when they are created.

procfilt.dat

This file defines which processes captured by Cloudpaging should be included when the captured installed is merged into the project file. If this file is blank, all processes are merged. If any entries are found, only those processes are captured and merged.

Each entry consists of the process executable name (SPOOLSV.EXE), followed by a TRUE (all new child processes spawned by the process will be captured and merged) or FALSE (all child processes spawned by the process will be ignored). Existing child processes are always ignored.

defprocsel.dat

This file controls the default settings for the Security screen of the Options dialog (see [Security Settings](#) on page 60). The process names and their settings in this file will be the initial settings for a new project.

Each entry in this file consists of the executable name for the process, some white space, TRUE or FALSE, more white space, and if necessary, TRUE or FALSE again. The TRUE/FALSE settings determine the initial security settings for the listed process.

The first TRUE or FALSE defines the SELECT-ability of level 4 files: can a process on a host PC detect the existence of file within the virtual namespace. The section TRUE or FALSE defines the readability: can the process actually access the contents of the file.

Example:

```
PROCESS1.EXETRUETRUE
```

```
PROCESS2.EXETRUEFALSE
```

PROCESS3.EXEFALSE

The first entry defines a process that can detect and read files in the virtualized appset, even if their disposition is virtual-isolated (level 4).

The second entry defines a process that can detect virtual files, even if their disposition is virtual-isolated (level 4), but not read them.

The third entry defines a process that cannot detect files in the virtualized appset, even if their disposition is virtual-integrated (level 3).

If the first value is FALSE, any second value will be ignored if present, because if a file cannot be selected, by definition it cannot be detected.

NOTE

This file is loaded once every time a new project is created. Subsequently, loaded information from this file will be saved with the new project. Reopening an existing project will not reload this file and information will be reloaded from the saved project. In other words, changes to this file will affect only future projects when they are created.

fileexcluded.dat

This configuration file defines which files and folders are excluded from **Cloudpaging Player's** normal sandboxing behavior (see [Sandboxing Settings](#) on page 58). The file paths listed here on the host PC will be accessible to the virtualized application, and changes would be written to the physical folders, rather than be sandboxed within **Cloudpaging Player's** virtual namespace.

Examples:

An absolute path: `C:\Abc\Xyz`

Special folder templates: `?documents?` for *My documents*

Environment variables: `%programfiles%\xyz`

By default, the personal document folders ("My Documents," "My Pictures," etc.) are excluded from sandboxing.

NOTE

This file is loaded once every time a new project is created. Subsequently, loaded information from this file will be saved with the new project. Reopening an existing project will not reload this file and information will be reloaded from the saved project. In other words, changes to this file will affect only future projects when they are created.

regexcluded.dat

This configuration file defines which registry keys are excluded from **Cloudpaging Player's** normal sandboxing behavior. (see [Sandboxing Settings](#) on page 58). The registry paths listed here on the host PC will be accessible to the virtualized application, and changes would be written to the local registry, rather than be sandboxed within **Cloudpaging Player's** virtual namespace.

By default, this file is empty and all registry keys are sandboxed.

NOTE

This file is loaded once every time a new project is created. Subsequently, loaded information from this file will be saved with the new project. Reopening an existing project will not reload this file and information will be reloaded from the saved project. In other words, changes to this file will affect only future projects when they are created.

regdifferences.dat

This file defines registry keys and registry values which will be compared before and after capturing the installation, keeping the difference between the original value and the new value. For example, if the original value was "abc" and after installation the value is "abcdef," only the "def" portion is written to the registry when virtualizing the application on a host PC.

Separate the registry key path and value name using a tab. If key path contains spaces, then enclose the path in quotes ("") and escape any backslash characters (\\).

rlnfilt.dat

This file defines registry key where the data is path names is stored in the registry as a short file name, which will be converted to the full long file name in the appset file.

For example, a folder named "Microsoft Office" might be shown as "MICROS~1" in a registry key. This might cause Player to confuse this folder with another folder on the Target PC, which may have assigned "MICROS~1" to a different long folder name, such as "Microsoft Silverlight."

File paths in the registry keys listed in his file will be converted to their long file name (LFN) equivalent.

appratng.dat

This file is used to define the local game ratings. By default, this file contains entries relevant to the United States. If using **Cloudpaging Studio** with games from another locality, you can edit this file to localize **Cloudpaging Studio** to the rating system in use in that country.

NOTE

This file is loaded once every time a new project is created. Subsequently, loaded information from this file will be saved with the new project. Reopening an existing project will not reload this file and information will be reloaded from the saved project. In other words, changes to this file will affect only future projects when they are created.

Appendix C: Command Line Usage

Syntax

Cloudpaging Studio can be used from the command line according to the following syntax:

```
JukeboxStudio [file_path | {-d [aib_file|tok_file] -f[:p|e|b|s|u|t|d][...] file_dump |
-e[:d|r][...] registry_dump} | {-x:f|r|e[...] [stp_file|tok_file] -t [extract_location]
} | {-p workspace_file -o output_file [-i [app_id]] [-n app_name] PREFETCH_OPTIONS}] [-l
log_file_override] [-h | -?]
```

PREFETCH_OPTIONS: [-m[:n] prefetch_file [...]] [-r[:n] prefetch_file [...]]

Commands

The various command line parameters are described in the table below:

file_path	Open specified file (.stw, .tok or .aib) in Cloudpaging Studio
-d [aib_file tok_file]	Dump file and registry entries from the specified .aib file or dump the setting (CAE attributes, virtualization settings, sandbox settings, and security settings) from the specified .tok file. The -f parameter must be used when dumping the contents of the .tok file.
-e[:d r] registry_dump	Write registry entry dump to specified file. Optionally include registry keys dispositions if "d" is specified or resolve templates if "r" is specified. Both options may be specified as in "-e:d,r," for instance.
-f[:p e b s u t d] file_dump	File to write the contents of the dump command with the "-e" or "-d tok_file" commands. Optionally for aib file: p - include full paths e - include extended information b - include paging block ID if 'b' s - include time stamps u - include times as UTC t - include tab separation d - include for display
-h, -?	Display command line help text.
-i app_id	Create appset using a new application ID. Use app_id if specified (a GUID in the form 12345678-90ab-cdef-1234-567890abcdef, with or without enclosing braces); otherwise generate a new application ID automatically.
-l log_file_override	Write log entries to specified file (overrides the use of the default log file).
-m[:n] prefetch_file	Merge content of specified file with stage-n prefetch in created appset, where n is either 1 or 2. If stage number n is not specified, 1 is assumed.
-n app_name	Create appset using a new application name.

<code>-o output_file</code>	Store created appset in specified .stp file.
<code>-p workspace_file</code>	Create appset specified by workspace (.stw) file.
<code>-r[:n] prefetch_file</code>	Replace stage-n prefetch in created appset with content of specified file, where n is either 1 or 2. If stage number n is not specified, 1 is assumed.
<code>-x[:f r e][stp_file tok_file] [-t extract_folder]</code>	<p>Extract the contents of the .stp or .tok file to the system. The f r e parameters determine what type of asset to extract: f - files and folders r - registry values and keys e - environment variables</p> <p>At least one parameter from f r e must be specified.</p> <p>The -t parameter specifies folder to extract the specified assets. Unless specified, files and folders will be extracted to the file system to their respective locations, and registry entries will be merged into the system registry.</p>

Notes:

- Items in braces { } are to be considered collectively as a group.
- Items or groups in square brackets [] are optional.
- An ellipsis in square brackets [...] indicates that the previous option may be repeated any number of times.
- Pipe characters (|) separate items or groups only one of which may be chosen.
- Forward slashes (/) may be used instead of dashes (-) as a switch prefix.
- Both -r and -m options may be specified simultaneously for the same stage number. In this case the prefetch is first replaced according to the -r option, then augmented according to the -m option.
- Options -r or -m may be specified multiple times for different stage numbers. If any of these options is specified more than once for the same stage number, only the last option specified for that stage number will apply.

Glossary

The definitions in this glossary are in the context of Cloudpaging Server and may not necessarily correspond to more general definitions.

Term	Definition
access token	<p>A file that is used to monitor the license agreement of a cloudified application, which is passed back and forth between Cloudpaging Player and Cloudpaging Server.</p> <p>When Cloudpaging Player receives an access token, permission is granted to run the application. If a user is running multiple or expired sessions, Cloudpaging Player may be denied an access token.</p>
activation	The first of six stages that launch an application from Cloudpaging Player.
admin	See system administrator
Admin service	The Cloudpaging Server component that provides configurable web-based interfaces for the administrator, as well as web interfaces for users to launch cloudified applications.
Application Installation Blueprint (AIB) file	An Application Installation Blueprint (AIB) file contains all the files' metadata and registry information for the appset. The extension for this file is .aib.
AppEvent	A Cloudpaging Studio process or action that uses a <i>trigger</i> to run an executable program or script. In turn, the script contains instructions to perform one or more specified runtime actions that are required by the application being cloudified, such as launching another application. AppEvent scripts and executable programs can be written in any scripting or programming language. AppEvents are also called <i>Configurable AppEvents (CAEs)</i> .
application	Any Windows software program or game, such as Microsoft Word or Adobe Acrobat. See also cloudified application .
application license policy	A policy that controls the conditions under which users can access and use an application. These conditions are defined by settings such as license type, total number seats, offline duration (fixed licenses only), etc.
appset	See cloudified application

artifacts	For Cloudpaging Server, artifacts are unwanted changes or additions, such as DLLs, that occur most often when "All processes" is selected during the application installation capture portion of the packaging process.
authentication	The process by which the system validates a user's logon information. The user's name and password are compared against an authorized list, and if the system identifies a match, access is granted to the extent specified in the permission list for the user.
cache	A temporary storage place (often a specified portion of computer disk space or RAM memory) for frequently needed data that can be retrieved quickly. See also Client cache.
Configurable AppEvents (CAE)	There is one CAE file per AppEvent trigger which generates a response such as running a VB script that installs other needed applications. See also AppEvent.
Common Gateway Interface (CGI) script	A script that uses the CGI (Common Gateway Interface) protocol for interfacing external application software with an information server, commonly a web server. In the case of a web server, the CGI script responds to requests from client web browsers by returning output. Each time a request is received, the server analyzes what the request asks for, and returns the appropriate output.
clean packaging PC	For Cloudpaging Studio, a PC that has the Windows operating system installed, as well as the most recent drivers (and possibly imaging software), but is free of any other unnecessary applications.
client	For Cloudpaging Server, a PC with Cloudpaging Player installed. In general, a client is a computer that accesses shared network resources provided by another computer, called a server. Also, a client can be an application or process that requests a service from a process or component. In a client-server environment, the workstation is usually the client computer.
cloudified application	<p>An application that has been converted to a cloudified format using Cloudpaging Studio. A user can access this program from any Windows computer with an Internet broadband or company intranet connection using Cloudpaging Server, without installing the program on a client PC.</p> <p>There are three different names for a "cloudified application" depending on the point of</p>

	<p>view (POV) of the person using it:</p> <ul style="list-style-type: none"> • From a Appset Designer's POV - "<i>appset</i>" • From a Cloudpaging Server admin's POV - "<i>application</i>" • From a user's POV - "<i>cloudified application</i>"
cloudifying	The process of converting a software application into a form that can be cloudpaged to users' PCs. See also cloudified application .
Cloudpaging	The proprietary protocols and technologies developed by Numecent which allow cloudified applications to be delivered from Cloudpaging Server to a client device on demand.
Cloudpaging database	The Cloudpaging Server component that tracks user information and server resources for a given enterprise. All system and user information is stored in this database, including system configuration information, topology, logs, application information, application licenses, usage, and user account information. See also Database Server .
Cloudpaging service	This is a general term that refers to an Cloudpaging Server service: <i>Paging service, License service, or Admin service</i> .
Configurable AppEvent	See AppEvent
database server	A machine that stores profile, usage, and system information of Cloudpaging Player users. The database server is a component of the Cloudpaging Server system. See also Cloudpaging database .
disposition layers	<p>The layers used to configure individual system resources, such as a file, folder, registry key, or registry value. The layer determines whether the system resource can be seen by the local system, and whether it is permanent or can be removed.</p> <p>Layer 1 (Installed-permanent) Copies assets (files, folder, registry keys, and registry values) permanently onto the local system, and can be seen by the entire local system.</p> <p>Layer 2 (Installed-temporary): Installs assets during the activation process, and uninstalls assets during the deactivation process. The original asset is backed up before the new asset is installed, and when the new asset is uninstalled, the original asset is restored.</p> <p>Layer 3 (Virtual-integrated): Assets that can be seen both by the virtualized application and the local system, but are not physically installed on the local system.</p>

	Layer 4 (Virtual Isolated): The default setting. Assets that can only be seen by the virtualized application, and are not physically installed on the local system. See also virtualization .
Data Source Name (DSN) file	This file is used in the process of linking a Cloudpaging database to a Microsoft Access database.
End User License Agreement (EULA)	This is typically an HTML file that consists of a legal agreement the user acknowledges regarding the warranties and conditions of use for a software program.
fixed license	An application license that permits access to the application when the user is working offline (not connected to the enterprise's network). <u>NOTE:</u> A fixed license is in use when an application is in the Ready or Running state. See also floating license .
floating license	An application license that requires the user's PC to be online (connected to the enterprise's network) in order to access the application. With this license the application cannot be accessed offline. <u>NOTE:</u> A floating license is only in use when an application is Running, but not when it is in the Ready state. See also fixed license .
Fully Qualified Domain Name (FQDN)	An unambiguous domain name that specifies the exact location in the Domain Name System's tree hierarchy through to a top-level domain, and finally to the root name server. Some applications, such as web browsers, will try to qualify the domain name part of a Uniform Resource Locator (URL) if the DNS resolver cannot find the domain. An FQDN differs from a regular domain name by its absoluteness; a default domain name will not be added.
Globally Unique Identifier (GUID)	16-byte code that identifies an interface to an object across all computers and networks. Such an identifier is unique because it contains a time stamp and a code based on the network address hard-wired on the host computer's LAN interface card. These identifiers are generated by a utility program.
fileshare	A folder and its contents that is shared with other users on a network.
firewall	A firewall is a software and/or hardware barrier that protects private and company information from

	external threats. For companies, all communication is routed through a proxy server outside of the organization's network, and the proxy server decides whether it is safe to let a particular message or file pass. The proxy server hides the true network addresses.
ICO file (application icon file)	This is the graphic file for the application icon provided by the software vendor (for example, the Word icon for each Word file).
keytab file	Used by Cloudpaging Server/Enterprise Portal, this file contains pairs of Kerberos principals and DES-encrypted keys (derived from the Kerberos protocol). The Keytab file can be used to log into Kerberos without being prompted for a password. The most common use of keytab files, however, is to allow scripts to authenticate to Kerberos without human interaction or store a password in a plaintext file.
license key	A key used by software products to protect against piracy. Typically a license key includes a digital signature that locks to a specific computer to ensure the software cannot be executed on other computers.
License Service	A Cloudpaging Server component that manages <i>application licenses</i> and meters application usage. The primary functions are to grant, renew, and delete access tokens, record application usage, and help with server <i>load balancing</i> .
load balancing	<p>The process of distributing processing and communications activity evenly across a computer network so that no single device is overwhelmed. Load balancing is especially important for networks for which it is difficult to predict the number of requests that will be issued to a server. If one server starts to get swamped, requests are forwarded to another server with more available capacity.</p> <p>Load balancing refers to a server cluster sharing information requests equally across all of its active nodes. This can be done either statically, by tying clients directly to different back-end servers, or dynamically by having each client tied to a different back-end server controlled by software or a hardware device.</p> <p>See also failover.</p>
local paging	The process used by Cloudpaging Player to cloudpage an appset from a file located on the Target

	PC. Local paging allows you to test several appset iterations quickly, without publishing the appset to Cloudpaging Server.
log file	A file that stores messages generated by an application, service, or operating system. These messages are used to track the operations performed.
machine	See server
network segment	A portion of a network separated by a Layer 3 switch.
node	For Network topology, equipment situated at a point of branching of physical connections, or terminating a physical connection.
Appset Designer	The person responsible for using Cloudpaging Studio to convert Windows-based applications into a cloudified format.
page	A fixed-size portion of an application that is fetched from the Cloudpaging Server system to Cloudpaging Player on users' PCs.
paging	The process of transferring a cloudified application from Cloudpaging Server to a user's PC.
persistent binding	A setting that allows network switching hardware to constantly route traffic between two computers so that a <i>session</i> is maintained. With load-balancing switches, or Layer 3 switches, persistent binding settings facilitate this constant route instead of using the most efficient route.
Player Cache	The segment of disk storage on the Cloudpaging Player PC that temporarily stores pages that are paged from an enterprise's Cloudpaging Server. See also cache .
Prefetching	The process of fetching application pages prior to their being explicitly requested by Cloudpaging Player, in anticipation of being needed by the application. Stage-1 Prefetch: This group of application pages are those that are needed to start the application, as well as to perform typical actions for the first few minutes of using the program, such as opening files, saving files, etc. Stage-2 Prefetch: This group of pre-cloudpaged data consists of application pages that are needed to

	accomplish other important tasks the user is likely to perform.
product license key	The license key that is required in order to start the Cloudpaging Server system and register users.
project file	See Project Workspace .
Project Workspace	The Cloudpaging Studio file (.stw) that stores all packaging configuration settings for an application that is being cloudified. This file is also known as the <i>project file</i> .
protocol	Set of rules and conventions for sending information over a network. These rules govern the content, format, timing, sequencing, and error control of messages exchanged among network devices.
realm	A database of user names and passwords that verifies valid users of a web service or application and displays a list of roles associated with each valid user.
Repository	The Cloudpaging Server folder where all cloudified applications are stored. This folder must be accessible by all other Cloudpaging Server components.
scaled topology	The installation of Cloudpaging Server components on multiple machines. For example, a multi-machine topology might include: <i>Paging</i> , <i>License</i> , and <i>Admin services</i> each installed on three separate machines, duplicates of these three machines for <i>load balancing</i> and <i>failover</i> ; a machine for the Cloudpaging database ; and (for Enterprise Portal) a machine for the Web Portal.
seats	The total number of users that can concurrently run an application that is authorized by a particular application license.
security	Techniques for ensuring that data stored in a computer, or transferred from one computer to another, cannot be read or compromised. Most security measures involve data encryption and passwords. Data encryption is the translation of data into a form that is unintelligible without a deciphering mechanism. This typically requires a password, which is a secret word or phrase that a user enters to gain access to a particular program or system.
symmetric key	See session key .
server	A computer (or "machine") onto which Cloudpaging Server components or other system components are installed.

service	For Cloudpaging Server, the sub-component of of the system that delivers application <i>pages</i> to <i>clients</i> and provides other Cloudpaging Server functions. See also Cloudpaging service and Windows service .
service interruption	An interruption in the delivery of pages from Cloudpaging Server. Cloudpaging Player handles and recovers from service interruptions that may occur during normal use. For example, Internet traffic can affect how data travels to and from networks.
session	An open line of communication between a user and another computer, or between two computers. A session can remain open even if no communication is on-going. This allows for the communication to continue at a later time. Once a session has been evicted, then the communication will no longer be allowed and a new session must be established.
session key	An encrypted key that is randomly generated to ensure the security of a communications session between a user and another computer or between two computers. Session keys are sometimes called symmetric keys, because the same key is used for both encryption and decryption. A session key can contain information about the user that is communicating.
signed certificates	Web sites use SSL certificates to secure communication with a web browser. A certificate is digitally signed by the creator and this signature is used by application software to validate the legitimacy of the certificate by others. Most web sites use a signed certificate issues from a certificate authority (CA), such as Thawte or Verisign.
STC file	See secure sockets layer .
STP file	This file contains all the data pages of the application being cloudified. This file is the largest portion of the STP file.
Paging service	The Cloudpaging Server component that stores, manages, and pages cloudified applications to Cloudpaging Player on user's PCs. The Paging service adheres to restrictions established by the <i>License service</i> . See also License Service .
strong password	A password that satisfies these basic guidelines to protect against password-guessing system attacks: <ul style="list-style-type: none"> • Include numbers, punctuation, and upper and lowercase letters in the password.

	<ul style="list-style-type: none"> • Use a password having at least eight characters. • Avoid a password based on repetition, dictionary words, letter or number sequences, user names, or biographical information such as names or dates.
system administrator	The person who manages the enterprise's Cloudpaging Server. In general, the person responsible for setting up and managing domain controllers or local computers and their user and group accounts, assigning passwords and permissions, and helping users with networking issues.
target application	An application that has been selected to be cloudified using Cloudpaging Studio.
templates	Aliases that map to actual Windows folder paths. For example, WindowsDir maps to c:\windows.
token file (TOK)	This file contains the type of license and other security data for the cloudified application. It also contains general information about the application, including location of servers and other paging sources, game rating, and operation system compatibility. The token file extension is .tok .
token renewal frequency	Frequency at which Cloudpaging Player must renew its access tokens in order to continue using Cloudpaging Server applications. Currently, the frequency is set at 10 minutes.
Tomcat	Apache Tomcat is a web container, or application server developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management, but can also be configured by editing configuration files that are normally XML-formatted. Tomcat includes its own internal HTTP server.
trigger	A process or condition that causes an action to start, such as running a VB script.
virtual machine (VM)	Software that mimics the performance of a hardware device, such as a program that allows applications written for an Intel processor to be run on a Motorola processor.

	A VM is a computer that does not exist as a physical device, but is simulated by another computer.
Virtualization	The process that occurs after activation and prefetch of an appset, which makes the application appear to the user's Windows system as if it were physically installed.
Web Portal	(For Enterprise Portal systems) A website that is integrated with Cloudpaging Player and Cloudpaging Server. Enterprise Portal is the web interface that functions as the front-end that an admin uses to configure and control all Cloudpaging services.
Web server	(For Enterprise Portal systems) The machine that contains the Web Portal.
Windows service	An application that starts when Windows is booted and runs in the background as long as Windows is running. Windows services can have special privileges that regular applications do not. For instance, on a limited user account a service could access certain parts of the Windows Registry that a regular application could not access. Most Windows services do not have a user interface and are loaded when Windows starts.
Windows Internet Name Service (WINS)	Microsoft's implementation of NetBIOS Name Service (NBNS), a name server and service for NetBIOS computer names. Effectively WINS is to NetBIOS names what DNS is to domain names - a central mapping of host names to network addresses. Like DNS, it is broken into two parts: (a) a Server Service that manages the encoded Jet Database, server to server replication, service requests (query & registration, renewals, de-registration), and conflicts; and (b) a TCP/IP Client component which manages the client's registration, renewal of names and takes care of queries.