

# Contents

1. Introducing CloneApp.....	3
1.1 What is CloneApp? .....	3
1.2 What can you use it for? .....	3
1.3 What makes CloneApp different from other Backup Software? .....	3
1.4 System requirements.....	3
1.5 About the Project.....	3
2. Using CloneApp .....	4
2.1 The User Interface.....	4
2.2 Backup and Restore Application settings.....	5
2.2.1 Backup Application settings .....	5
2.2.1 Restore Application settings .....	5
2.2.3 Background Information about Backup and Restoration Process .....	5
2.2.4 How can I Restore a Backup to another Computer?.....	6
2.3 Importing new Applications .....	6
2.4 Checking for CloneApp updates.....	6
3. CloneApp Rules.....	7
3.1 Structure of a Plug-in file.....	7
3.2 Supported Variables .....	8
3.3 Supported Commands .....	9
3.3.1 Backup Files .....	9
3.3.2 Backup Folders .....	9
3.3.3 Backup Registry Keys.....	10
3.3.4 Run Files .....	10
3.3.5 Run Commands.....	10
3.3.6 Kill Tasks.....	11
3.3.7 Detect .....	11
3.4 External Plug-ins .....	11
3.4.1 Example of an External Plug-in .....	11
3.4.2 Supported External Plug-ins.....	11
4. CloneApp Settings .....	12
4.1 Clone Path.....	12
4.2 Log Path .....	12

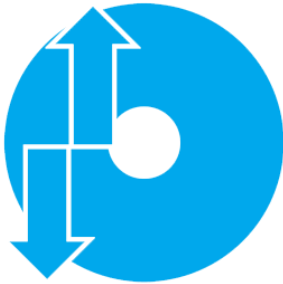
## DOCUMENTATION CLONEAPP

4.3 Editor Path .....	12
4.3.1 Internal Editor.....	12
4.4 7z Compression.....	13
4.5 Clone Modes .....	13
4.5.1 Clone Mode: Clone Apps in same or separate folder: .....	13
4.5.2 Confirmation Mode: Display or Hide Clone conflicts.....	13
4.6 Theme Mode .....	14
4.7 Configuration files .....	14
5. Advanced Usage .....	15
5.1 Including Custom Items to the Backup process .....	15
5.2 Log operations .....	15
5.3 Command-line parameters.....	16
5.4 Backup Plug-ins and work with certain Apps .....	17
6. Troubleshooting.....	17
7. Documentation Info .....	17

**PLEASE NOTE that the information in this Documentation always refer to the most recent Version of CloneApp, which can be downloaded [here](#)**

# 1. Introducing CloneApp

---



## 1.1 What is CloneApp?

CloneApp is a small, **fully portable** utility for computers running Microsoft Windows that is designed to easy backup and restore Application settings and Windows configurations that are stored in Windows directories, profile folders and Registry.

## 1.2 What can you use it for?

As you can guess from its Name **CloneApp**, it is used to copy and restore configuration files from Applications and Windows components.

E.g. CloneApp can be helpful keeping Windows and program settings when reinstalling Windows or just for Backup purposes for not losing the Configurations, when reinstalling Applications.

## 1.3 What makes CloneApp different from other Backup Software?

With CloneApp you can create Application backups of a different kind. Instead of completely backing up all files of a program or even a complete Windows partition etc., CloneApp will Backup ONLY the Configuration files of an Application itself from Windows directories and Registry.

This is done quickly and saves a lot of space.

**Info:** In many ways, backing up ONLY the Configuration files is also the more secure way. E.g. if your System becomes insecure due to malware or mistake or unstable a fresh install is sometimes the only way to go.

## 1.4 System requirements

CloneApp will run on any PC running **Microsoft Windows 7, Windows 8, Windows 8.1, Windows Server 2008-2012, Windows 10** (all editions, including 64-bit)

Since CloneApp is fully portable and has a very low memory usage, there are no minimum memory or hard drive requirements. It is not bonded on 3<sup>rd</sup>-Party Libraries or has other Dependencies and it does not write anything to the Registry.

## 1.5 About the Project

The Application is very new on the market (initial release was 05. May 2015) and some Antivirus-Engines could maybe trigger false positives. Also it is currently NOT digitally signed. Since this is a one-man Project, it relies on Donations. **If you want to contribute and Support the Development of CloneApp, you can send me a Donation [here](#).**

## DOCUMENTATION CLONEAPP

If you enjoy CloneApp please introduce it your beloved and more number of people, write about CloneApp on your Blog and help me grow CloneApp.

**Note: CloneApp contains NO spyware or adware.**

## 2. Using CloneApp

---

### 2.1 The User Interface

The User Interface (UI) is separated into four tabs. The Main tab is captioned with the Name **Clone**. This is also the first tab you will see, when starting CloneApp.

The **Clone** tab includes three areas. On the **left** hand you have the **Supported Apps/Plug-ins list** (*What exactly Plug-ins are can be read in the chapter **CloneApp Rules***), which is listing all supported Applications in CloneApp. Next to the Apps list, also a **Installed** list is available. This one is listing all Installed Applications on your running System (*Read more about this feature in the chapter **Advanced Usage***)

On the **right** hand, you have the **Description fields**, which display some Information about the selected Plug-in like Name of the Plug-in, Information what the Plug-in script does and the **Author of the Plug-in**.

Next to the Description field, you will also see a Button captioned with the Name **What is being backed up?** When clicking on this Button, CloneApp will analyze the Files, Folders and Registry Keys for their existence belonging to the Applications and Windows features you have selected. This may take some time.

**Note: The **What is being backed up?** feature is **ONLY for Preview purposes**. No files are backed up during the Process.**

Next to the **What is being backed up?** Button you will see an **Edit** Button, which can be used to open the Plug-in file itself for showing the Script behind the selected Plug-in. The Plug-in file is opened by default with Windows **Notepad** (*Read more about this feature in the chapter **CloneApp Settings***)

Beneath the **What is being backed up?** Button you will also discover a small **summary of the Plug-in Script**, which counts the Number of backed up Files, Folders, Registry Keys and optional Commands of the selected Plug-in.

The Bottom of the Main UI is showing the **Status-, Log window**. This window logs all actions during the Backup and Restoration process. (*Read more about this feature in the chapter **Advanced Usage > Logging***)

## 2.2 Backup and Restore Application settings

### 2.2.1 Backup Application settings

**Note:** You should be **running CloneApp always with Administrator rights**. You can still use it as a normal or restricted user, but Windows may prevent you from copying certain files.

To start a Backup process click on the left Navigation Icon **Clone**.

Choose now the Apps you want to be backed up by selecting their check boxes. If you are unsure which Apps to select, you can also use the feature **Select Installed**.

This feature works very precisely, checks every deposited Configuration file in the Plug-in files for their existence, and selects all matches.

For **starting a Backup**, click the **Start CloneApp** button. CloneApp starts now immediately the Backup process and exports all files to the Clone Path defined under Options (*Read more about this feature in the chapter **CloneApp Settings***). The Backup process itself is straightforward. It runs the process for each Application that you have selected individually and echos the progress that it makes in the Status-, Log window.

### 2.2.1 Restore Application settings

The same procedure from the Backup process is also used in the Restoration process, where you only have to use the **Restore** button. **The Basis for the Restoration process is the Plug-in file itself and the Restoration file `cloneapp2.ini`**, which is being created after each Backup Process and where all relevant Settings are saved like the ClonePath, Clone Mode and the Confirmation Mode (*Read more about this feature in the chapter **CloneApp Settings > Clone Modes***).

### 2.2.3 Background Information about Backup and Restoration Process

CloneApp runs the Backup and Restoration process very careful.

Dialogs may be displayed by default to you during the process, for instance, when a file cannot be copied, because when an application is still running during the process or when folders cannot be copied because the main folders do not exist since the application is not installed. In the last case, CloneApp will ask you whether the folders should be created. This could also happen, when an Environment Variable does not exist and CloneApp does not know where to copy and/or restore the Files.

Read more in the chapter **CloneApp Settings > Clone Modes and how to handle Dialog boxes and Error Messages in CloneApp**.

CloneApp will log every action taken in the Status window.

## DOCUMENTATION CLONEAPP

### 2.2.4 How can I Restore a Backup to another Computer?

Just copy the CloneApp program folder including the CloneApp Plug-ins and of course the Backup folder including the `cloneapp2.ini` to the Machine you want to restore the Configuration.

CloneApp writes the Path where the files has been backed up (Clone Path) also to the `clonapp2.ini`

The Username on the target Machine you want to restore the Configuration could maybe be another as on the Source Machine where the Backup has been made. In this case CloneApp could maybe return the Error message "**Restoration Settings could not be loaded. Clone Path has been reseted to Default. Please try again**". In the second attempt the Clone Path in `cloneapp2.ini` will be prepared and reseted to the default Clone Path registered in the Options tab.

### 2.3 Importing new Applications

New Plug-ins for CloneApp can be found [here](#).

To import and install a new downloaded Plug-in use the **Import Plug-in** feature (Clone tab) of CloneApp. *(Read more about this feature in the chapter **CloneApp Rules**)*

### 2.4 Checking for CloneApp updates

The **Check for updates** feature checks whether a more recent version of CloneApp is available to download. To use the feature, your computer must have a connection to the internet. If your internet connection uses a proxy server, make sure your web browser connection settings are configured correctly.

To check for updates go to the **Get Started** tab, click the **Check for updates** link next to the Name label CloneApp. The Version label beneath will tell you if you have the latest version, or will pop-up a download Message box if a newer Version is available.

## 3. CloneApp Rules

---

CloneApp is based on an .INI Engine to determine which Applications and Windows configuration files to Backup and/or Restore and how to do that. .INI files are simple text files with plain text, so you can edit them with Notepad. CloneApp brings by default already pre-configured Plug-ins for dozen popular Applications. To extend CloneApp's functionality you can create your own Plug-ins.

### 3.1 Structure of a Plug-in file

#### [Info]

**Title**=Name of the Plug-in. Title is showed in the UI > Name field

**Version**=Shows with which Version a Plug-in is compatible (Optional)

**Description**=Shows a short Description of what is being backed up

**Author**=Mirinsoft or <http://www.mirinsoft.com>

**AuthorURL**=Optional in URL format: <http://www.mirinsoft.com>

**Warning**=Display Warnings (Optional)

**BackupFile**=1

**BackupFolder**=1

**BackupRegKey**=0

**RunCommand**=0

#### [Files]

**File1**=**BackupFile**|%AppData%\Example App\Configuration\config.ini

**Info:** The last **[Info] commands** (*BackupFile, BackupFolder, BackupRegKey, RunCommand*) summarizes **ONLY** how many Entries are processed during the Backup and Restoration process.

The **Execution process of a Plug-in** is done in the **[Files]** section.

The File entries must start at 1 and proceed sequentially upwards **File1** (File2, File3 etc.) followed by the **equality sign (=)**

Then the **Command** has to be added e.g. **RunFile** followed by the Path to the File/Folder/Registry Keys or Command.

**Info:** The Path entries are **NOT** case-insensitive.

## DOCUMENTATION CLONEAPP

### 3.2 Supported Variables

**CloneApp works with the Environment Variables of Windows.** The most common Variables like `%ProgramFiles%`, `%AppData%` etc. are defined on every Windows version. To see all Environment Variables configured on your System do the following.

*Start > Run > cmd > Type the command **set***

The "%" (**percent sign**) defines **Windows Environment Variables**.

Here is a list of the most common Variables on **Windows Vista and later** systems.

Variable	Place
<code>%ALLUSERSPROFILE%</code>	C:\ProgramData
<code>%APPDATA%</code>	C:\Users\{username}\AppData\Roaming
<code>%LOCALAPPDATA%</code>	C:\Users\{username}\AppData\Local
<code>%PROGRAMDATA%</code>	C:\ProgramData
<code>%PROGRAMFILES%</code>	C:\Program Files
<code>%PROGRAMFILES(X86)%</code>	C:\Program Files (x86) (only in 64-bit version)
<code>%ProgramW6432%</code>	C:\Program Files (only in 64-bit version)
<code>%PUBLIC%</code>	C:\Users\Public
<code>%SystemDrive%</code>	C:
<code>%SystemRoot%</code>	C:\Windows
<code>%USERPROFILE%</code>	C:\Users\{username}
<code>%WINDIR%</code>	C:\Windows

The "\$" (**dollar sign**) defines **Custom/Internal Variables of CloneApp** based on Windows API.

Variable	Place
<code>\$AppClonePath\$</code>	Clone/Backup/Output folder of CloneApp
<code>\$Date\$</code>	Date stamp e.g. in Clone Path
<code>\$Documents\$</code>	C:\Users\{username}\Documents
<code>\$AppData\$</code>	C:\Users\{username}\AppData\Roaming
<code>\$LocalAppData\$</code>	C:\Users\{username}\AppData\Local\Roaming
<code>\$ProgramFiles\$ (only 32-bit version)</code>	C:\Program Files (x86)
<code>\$FirefoxProfile\$</code>	Default Firefox Profile in AppData\Roaming

You can also define **Custom Variables in a CloneApp Plug-in file** e.g.

#### [Variables]

```
MyDocuments=%UserProfile%\Documents
```

The **Execution process of Custom Variables** is also done in the **[Files]** section.



## DOCUMENTATION CLONEAPP

### [Files]

```
File1=BackupFile|%MyDocuments%\config.ini
```

## 3.3 Supported Commands

CloneApp's core is the customizable Plug-in engine. This supports several Commands which will be explained in this chapter.

**Note:** All below used **Commands are case-insensitive.**

### 3.3.1 Backup Files

#### BackupFile or BackupFile64

```
File1=BackupFile|%AppData%\Example App\Configuration\config.ini
```

```
File2=BackupFile|%AppData%\Example App\Configuration\*.*
```

```
File3=BackupFile|%AppData%\Example App\Configuration\*.ini
```

```
File4=BackupFile|%AppData%\Example App\Configuration\*config*
```

The command used is **BackupFile**, which exports the File **config.ini** to the **Clone Path defined under the Options tab** in CloneApp. (Read more about this feature in the chapter **CloneApp Settings**). It supports also Filetypes/Wildcards, here \*.\* In the second example **File2** it would export all Files in the Configuration folder. You can also specify the Wildcard e.g. **File3 \*.ini** to export ONLY .INI-Files.

**Note:** The x64-bit Variant of the BackupFile Command, has to be used when working with x64-bit Environment Variables e.g. %ProgramW6432%.

**Info:** This command supports the Attributes write-only and hidden.

### 3.3.2 Backup Folders

#### BackupFolder or BackupFolder64

```
File1=BackupFolder|%AppData%\Example App\Configuration
```

This command is used to backup ONLY **complete** folders.

You can also define custom Output folders of your Choice, when working with this Command. E.g.

```
File2=BackupFolder|%AppData%\Example App\Configuration|OutputFolderNew
```

To eliminate redundancies, when backing up an Application which has different locations but same folder name, this command splits the absolute path of each backed up folder e.g.

#### Windows Themes

```
%LocalAppData%\Microsoft\Windows\Themes
```

The created Output folder **Windows Themes** creates five other Subfolders beginning with **%LocalAppData% \ Microsoft \ Windows \ Themes \ Themes**

## DOCUMENTATION CLONEAPP

Here the same, where instead of `%LocalAppData%`, the folder `%AppData%` will be created.  
**%AppData% \ Microsoft \ Windows \ Themes \ Themes**

**Note:** The x64-bit Variant of the BackupFolder Command, has to be used when working with x64-bit Environment Variables e.g. `%ProgramW6432%`.

**Info:** This command supports the Attributes write-only and hidden.

### 3.3.3 Backup Registry Keys

#### **BackupRegKey**

File1=**BackupRegKey**| `HKEY_CURRENT_USER\Software\ExampleApp`  
 or short Root RegKey format, i.e.

File2=**BackupRegKey**| `HKCU\Software\ExampleApp`

This Command creates also for each part of the Registry Key a separate subfolder to eliminate redundancies when backing up Registry Keys with identical Filenames e.g.

#### **7-Zip**

`HKEY_CURRENT_USER\SOFTWARE\7-Zip`

The created Output folder **7-Zip** creates three other Subfolders beginning with `HKEY_CURRENT_USER \ Software \ 7-Zip`

Here the same, where instead of `HKEY_CURRENT_USER` the folder `HKEY_LOCAL_MACHINE` will be created.

`HKEY_LOCAL_MACHINE \ Software \ 7-Zip`

This Command supports also the **creation of Custom Output folders**

File1=**BackupRegKey**|`HKEY_CURRENT_USER\Software\ExampleApp`|`HKEY_CURRENT_CustomFolder`

### 3.3.4 Run Files

#### **RunFile**

File1=**RunFile**|`%AppData%\Example App\Configuration\My Software.exe?`

Don't forget the "?" sign, which **executes the RunFile** command.

### 3.3.5 Run Commands

#### **RunCommand**

File1=**RunCommand**|`netsh advfirewall export $AppClonePath$WindowsFirewallSettings.wfw"`

This Command executes the command-line `netsh advfirewall` and exports the Windows Firewall Settings to the Backup directory of CloneApp.

## DOCUMENTATION CLONEAPP

The **RunCommand** supports the internal Variable **\$AppClonePath\$**. If enabled, it uses the default Clone Path directory of CloneApp.

### 3.3.6 Kill Tasks

#### **TaskKill**

```
File1=TaskKill|firefox.exe|WARNING
```

This Command is used to terminate tasks by process id (PID) or image name. Optional a hint can be showed by registering the **WARNING** command.

### 3.3.7 Detect

#### **Detect**

```
File1=Detect|HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\Word
```

**OR** (*ONLY one **Detect** Command is allowed*)

```
File2=Detect|%AppData%\Microsoft\Word
```

By default the Detection routine checks every deposited File, Folder and Registry Key in a Plug-in file for their existence. Optional you can use the **Detect** Command, which is been preferred if added, and check only the File, Folder and Registry Key existence in this Command.

## 3.4 External Plug-ins

CloneApp supports since **Version 1.04** also some command-line options for external Applications.

External Plug-ins are based on the default Plug-in file and the external Application itself, which has both to be imported to the Plug-in directory of CloneApp.

The external Plug-ins repository can be found [here](#).

### 3.4.1 Example of an External Plug-in

#### **ExternalPlug**

```
File1= ExternalPlug|Plug-ins\ProduKey\ProduKey.exe /stext $AppClonePath$\Product keys.txt
```

The Command used is **ExternalPlug**, which executes **NirSoft ProduKey** and exports the Product keys in a Text File. As destination path, the internal **\$AppClonePath\$** Variable is used.

### 3.4.2 Supported External Plug-ins

[Nirsoft ProduKey](#), which exports Product keys of Applications and Windows e.g. Windows, Office, Adobe products etc.

## 4. CloneApp Settings

---

In the **Options** tab, you will find some customizing settings for CloneApp to let it work the way you need it to.

### 4.1 Clone Path

The most important setting is the Backup destination path, called **Clone Path** where the backed up files are going to be copied and restored. **CloneApp automatically creates the Output/Backup directories.**

You can also use an **Variable/Dynamic Clone Path** e.g. `%UserProfile%\Desktop\CloneApp (Backup)-%Date%`

As you see, the Clone Path supports also the Variable **%Date%**, in the Format YYYY-MM-DD.

### 4.2 Log Path

In the Options tab, you will be also able to define **Log Path** of CloneApp where all actions performed during a Backup and/or Restoration process, are going to be saved. After each Backup and/or Restoration process, CloneApp fades-in a **Save** Button. The Log Path tells the Save Button where to export the generated Logs.

**Note:** You can also use a **Variable/Dynamic Log Path** e.g. `%UserProfile%\Desktop\`

### 4.3 Editor Path

This Path allows you to define a default Plug-in Editor e.g. Notepad++ By default Windows Notepad is deposited. All Plug-ins opened through the feature **What is being backed up?** in the **Clone** tab of CloneApp will be opened with the Editor defined in this path.

**Note:** You can also use a **Variable/Dynamic Editor Path** e.g. `%ProgramFiles%\Notepad++\notepad++.exe`

#### 4.3.1 Internal Editor

CloneApp brings also a small and fast built-in Plug-in Editor, which is enabled by default.

To make sure that it is enabled, please check for the Variable **\$AppEditor\$** entered in the Editor path. The internal AppEditor allows you to easily edit available Plug-in files and also to save new Plug-in files based on your editing.

## DOCUMENTATION CLONEAPP

### 4.4 7z Compression

**7-Zip** is an effective compression program. The 7z.exe program is used to compress, extract and update files through the command line. It provides superior compression. It is open-source. This makes it easy to obtain and also use it in CloneApp.

If Installed CloneApp will automatically detect the Installation path and recommend command line options by clicking the **Browse** Button, otherwise a Dialog will be displayed for entering a Custom path.

Use the **Compress** and **Decompress** Textboxes to edit the command line options for compression and decompression.

### 4.5 Clone Modes

#### 4.5.1 Clone Mode: Clone Apps in same or separate folder:

This option is set by default and let CloneApp **export all backed up files into one and same directory** which is defined under the Clone Path.

If you make a second click on this option, you will enable the second mode **Clone Apps in separate folder**. When using this option, CloneApp will create for each backed up Application a separate directory. The naming of the folders will be on basis of each Plug-in file.

If the Plug-in file is called **Windows Favorites** but the Application being backed up carries the name **Favorites**, then CloneApp will create a folder with the Name Windows Favorites and export the Favorites folder itself as a sub folder of Windows Favorites

#### 4.5.2 Confirmation Mode: Display or Hide Clone conflicts

This option is set by default and **displays confirmation dialogs** if a file/folder already exists and should be overwritten, if a folder does not exist and should be created or if the Application being backed up is running and the files cannot be copied. This and such other "normal" Windows file conflicts will always be displayed if an error occurs.

If you make a second click on this option, you will enable the second mode **Respond silent to all Clone conflicts**. While this option is enabled, CloneApp will respond to all Clone and Windows conflicts automatically with "Yes" and hide all User Interface and dialog boxes if an error occurs. In this case, file/folders will be automatically overwritten, folders will be created if they do not exist and running Applications/Processes will be ignored.

**Note:** By enabling this option, you will have to try NOT to evocate Windows conflicts e.g. a running Application such as Mozilla Firefox will not be backed up and/or restored completely while it is opened, because CloneApp (*and also every other Application/Process*) cannot copy and/or replace files in use.

## DOCUMENTATION CLONEAPP

### 4.6 Theme Mode

One other option, which is not to be found in the **Options** tab, but lets you modify the Appearance of CloneApp, is the **Theme Mode** feature. Right at the bottom of the CloneApp UI, you will find a small Button, called **night mode**, which is displayed in the default **day mode**. You can switch between these two modes to save your eyes at night.

### 4.7 Configuration files

Since CloneApp is a fully Portable Application, it saves its own settings in .INI files, which will be stored in CloneApp program **Data** folder.

Next to .INI files, also some other Configuration files are created in the program folder of CloneApp and which one, you will find out here:

#### **cloneapp.ini**

This is the main Configuration file, where all internal Settings from the **Options** tab of Clone App are stored.

#### **cloneapp2.ini**

This file is not available, until you run a Backup Process with CloneApp. After a Backup Process has been runned, this file will be automatically created, where all Settings are saved with which the Backup process has been taken. A copy of this file is also available in the folder where the files have been backed up. When starting a Restoration process, CloneApp will copy the **cloneapp2.ini** from the Clone Path to the **Data** folder of CloneApp and load the Configuration from it, which will be the basis for the Restoration process.

#### **plug-ins.ini**

This file stores the Status (enabled/disabled) of the App/Plug-in list available in CloneApp.

#### **CustomFiles.dat**

As you can guess from its Name, this file is used to save the Custom Files added in the **Custom** tab of CloneApp.

#### **CustomFolders.dat**

This file is used to save the Custom Folders added in the **Custom** tab of CloneApp.

#### **CustomRegKeys.dat**

This file is used to save the Custom Registry Keys added in the **Custom** tab of CloneApp.

#### **CustomCommands.dat**

This file is used to save the Custom Commands added in the **Custom** tab of CloneApp.

## 5. Advanced Usage

### 5.1 Including Custom Items to the Backup process

You can choose **specific Files, Folders and Registry Keys** and **Commands** to be included in the CloneApp Backup operations.

**Note:** User-defined Files are currently NOT included to the Restoration process.

To include Custom items to the Backup process, go to the **Custom** tab of CloneApp and select the items in that you want to be included to the Backup process.

The Custom Backup process supports the **same Filetypes and Environment Variables** as the default Application Backup. *(Read more about this feature in the chapter **CloneApp Rules**)*

Each Item category has to be separately proceeded by clicking the **Start Backup** button.

You can also **trigger the Custom section through the Plug-in Engine**, by using the following Commands:

Backup	Restore
<b>Backup Files</b> File1=BackupFileCustom	<b>Restore Files</b> File1=RestoreFileCustom
<b>Backup Folders</b> File1=BackupFolderCustom	<b>Restore Folders</b> File1=RestoreFolderCustom
<b>Backup Registry Keys</b> File1=BackupRegKeyCustom	<b>Restore Registry Keys</b> File1=RestoreRegKeyCustom
<b>Backup Commands</b> File1=RunCommandCustom	<b>Restore Commands</b> File1=RestoreCommandCustom

### 5.2 Log operations

CloneApp is logging every File, Folder, Registry and Command operation in the Status- and Log window. The Log output is quite restrained and marked only with an **<OK>** or **<Error>**

**Successful operations are marked with <OK>**

**Faulty operations are marked with <Error>**

There are **three scenarios** where you can get an **<Error>**

1. File, Folder or Registry Key does not exist (for Backup and Restoration purposes)
2. You do not have Administrator rights to access the File, Folder or Registry Key
3. The Application being backed up is running or an associated process.

## DOCUMENTATION CLONEAPP

### 5.3 Command-line parameters

You can use command-line parameters to change how CloneApp runs.

There are six parameters you can use when running CloneApp, as follows.

You can use them in batch files, scripting, shortcuts, or the Task Scheduler to automate CloneApp operations.

Parameter	Explanation
CloneApp.exe /AUTO	Detect installed Apps automatically and run Backup
CloneApp.exe /B	Run automatically Backup of selected Applications while showing CloneApp UI
CloneApp.exe /SB	Run silently Backup of selected Applications without showing CloneApp UI and ONLY an Status window on the bottom right corner of Windows
CloneApp.exe /CB "path_to_backup_folder"	Run silently Backup of selected Applications to Custom folder e.g. /CB "%UserProfile%\Dropbox\CloneApp-Backup\"
CloneApp.exe /R	Run automatically Restoration of selected Applications while showing CloneApp UI
CloneApp.exe /SR	Run silently Restoration of selected Applications without showing CloneApp UI and ONLY an Status window on the bottom right corner of Windows
CloneApp.exe /CR "path_to_backup_folder"	Run silently Restoration of selected Applications from Custom folder e.g. /CB "%UserProfile%\Dropbox\CloneApp-Backup\"
CloneApp.exe /P	Run Preview of selected Applications for Backup and Export them as a Text-File to the directory defined under the Log Path > Options menu



## DOCUMENTATION CLONEAPP

### 5.4 Backup Plug-ins and work with certain Apps

CloneApp offers numerous Plug-ins files for numerous Windows Applications. You will certainly not need every of those Plug-in files, because you do not have installed every Windows Application supported by CloneApp.

CloneApp brings the function to **Select Installed** Applications. This does not mean that only Installed Applications will be shown in the **Apps** window, when enabled.

To get a better Overview and to show only certain or your own Installed Applications, you can do the following.

1. Goto **Clone tab**
2. Right-Click on **Apps** window
3. Switch to Menu **Advanced**
4. Enable option **Backup Plug-ins**
5. Restart CloneApp

CloneApp will copy now all your selected Applications in the separate **My Apps** directory.

**Note:** CloneApp will always load the **My Apps** folder first.

To load the default Plug-ins directory, you just have to remove the **My Apps** folder.

## 6. Troubleshooting

Currently there are NO issues known while running CloneApp.

If you have Problems and need support, just follow CloneApp on the [Community forum](#), [Facebook](#) or [Twitter](#).

## 7. Documentation Info

Last Update on 02/20/2016