



Windows Client Management AG
Software Deployment Solutions

Scripting Framework PowerShell Toolkit

Installation of a Workplace for Packaging and Testing





1	Installation of the Scripting Framework Workplace.....	3
1.1	Completing the Installation.....	3
1.2	Introduction to Scripting Framework.....	6
1.2.1	Start Menu and Desktop Shortcuts	6
1.2.2	Scripting Framework Suite	8
1.2.3	Workplace Folder Structure.....	9
1.2.3.1	Configuration Files (C:\Scripting_Framework\Config)	9
1.2.3.2	Software Packages (C:\Scripting_Framework\Packages)	10
1.2.3.3	Scripting Framework Suite.....	11
1.2.3.4	Scripting Framework Engine	11
1.2.4	Scripting Framework Registry.....	14
1.2.5	Structure of the Scripting Framework Packages.....	15
1.2.6	Installing and Uninstalling Packages Manually	17
1.2.7	Creating Packages	17
1.2.8	Scripting Framework Functions	20
1.3	Closing Remarks	21



1 Installation of the Scripting Framework Workplace

This quick guide gives you an optimal overview of Scripting Framework. Please read the Manual for more detailed information.

Quickly and easily install a complete Workplace for packaging and testing software packages on your packaging computer. Network shares are not required, nor is a software distribution system.

Requirements

- Windows 64bit operating system
- A system restart may be necessary.

1.1 Completing the Installation

Run the file WinCM_Scripting_Framework_Workplace_Install_x64.exe as an administrator.


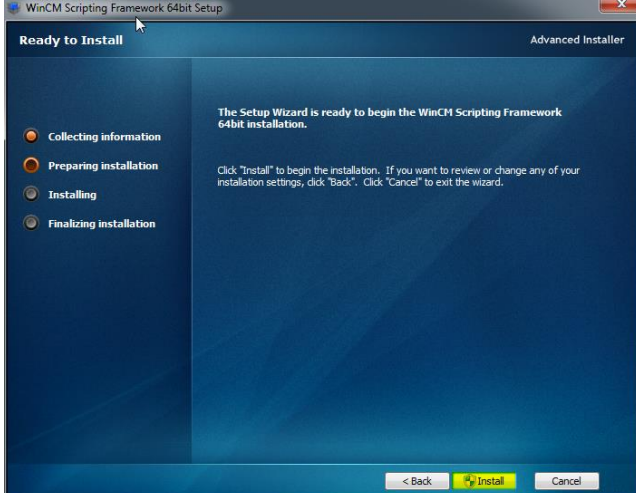
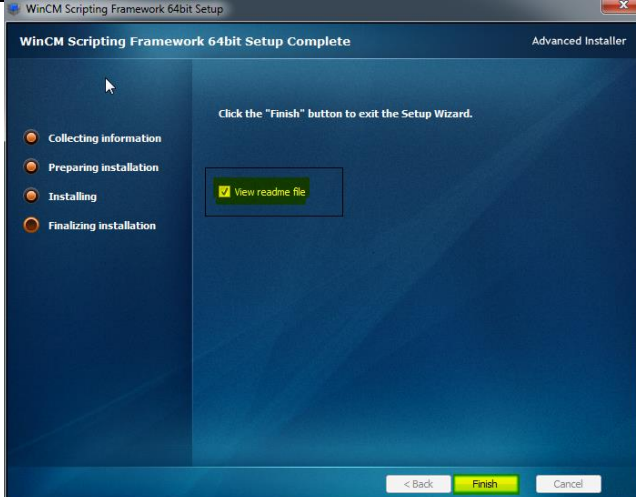
If necessary,

- Microsoft .Net Framework 4.5
- PowerShell 4.0

will be installed as well as Scripting Framework with Suite and several example packages.

Installing the Scripting Framework Workplace	
	<p>Next</p>
	<p>Scripting Framework requires at least .Net Framework 4 and Powershell 3.0 You can select or deselect the installation of the prerequisites.</p> <p>Next</p>



	Next
	Install
	<p>View the <i>readme file</i>. This contains important information about where to copy the license files.</p>



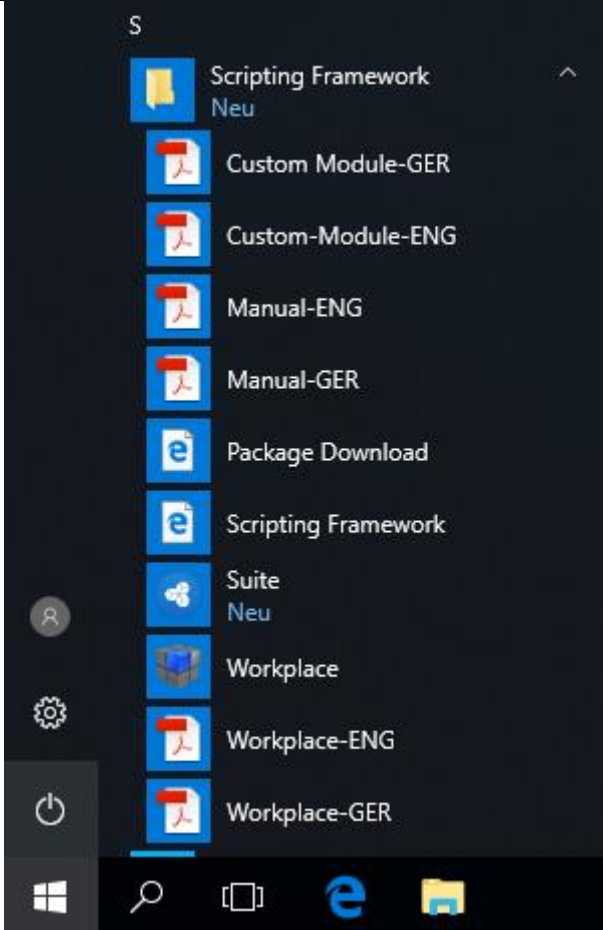
	<p>Instructions for the license files</p>
	<p>Please allow the system to restart (preferably after copying the license files) so that the installation of PowerShell and .Net Framework can be completed.</p>
<p>Copying the License Files</p>	
	<p>Copy the license file to the Ultimate Packager folder.</p> <p><i>C:\Scripting_Framework\Suite\Ultimate Packager</i></p>
	<p>Copy the license file to the Scripting Framework folder.</p> <p><i>C:\Windows_ScriptingFramework\Module</i></p>






1.2 Introduction to Scripting Framework

This overview will help you quickly become familiar with the *Scripting Framework Workplace*.

1.2.1 Start Menu and Desktop Shortcuts

 <p>The screenshot shows the Windows Start Menu with the 'Scripting Framework' folder expanded. The folder contains the following items:</p> <ul style="list-style-type: none"> Scripting Framework Neu Custom Module-GER Custom-Module-ENG Manual-ENG Manual-GER Package Download Scripting Framework Suite Neu Workplace Workplace-ENG Workplace-GER 	<p>The Start Menu now contains the folder Scripting Framework. This contains all necessary shortcuts.</p> <p>Custom Module-GER -> Instructions to create your own functions (German)</p> <p>Custom Module-ENG -> Instructions to create your own functions (English)</p> <p>Manual-ENG -> Scripting Framework Manual (English)</p> <p>Manual-GER -> Scripting Framework Manual (German)</p> <p>Package Download -> Internet link for downloading additional example packages</p> <p>Scripting Framework -> Scripting Framework product homepage with all the necessary information</p> <p>Suite -> Scripting Framework Suite – all tools are summarized here</p> <p>Workplace -> Scripting Framework file structure with Suite, example packages and configuration files (C:\Scripting_Framework)</p> <p>Workplace-ENG -> This document (English)</p> <p>Workplace-GER -> This document (German)</p>
---	---

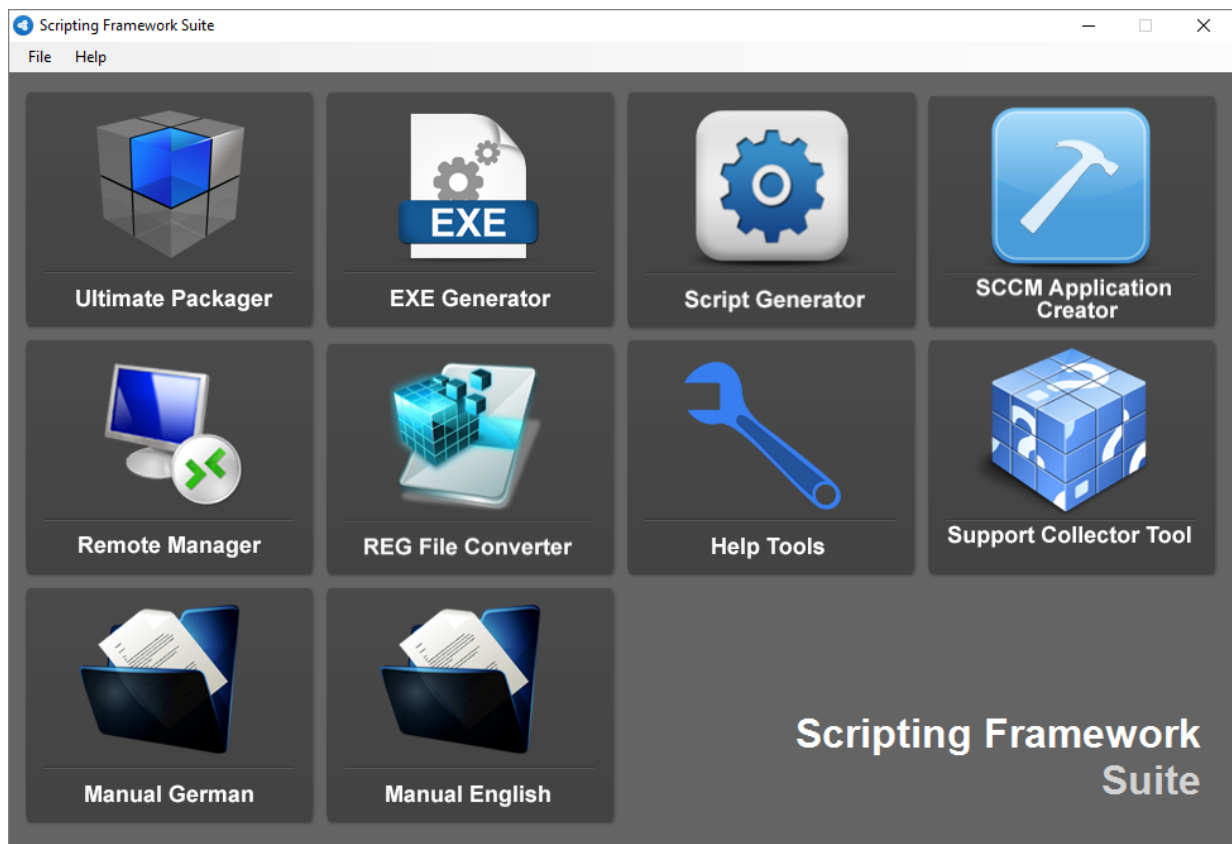


 PowerShell ISE  Scripting Framework Suite  Workplace		<p>The four most important shortcuts are added to the desktop (see description above).</p>
---	--	--



1.2.2 Scripting Framework Suite

In the *Scripting Framework Suite*, various tools are summarized.



You also have the option to create additional shortcuts for all modules or the Manual on the desktop, for example.

Ultimate Packager	C:\Scripting_Framework\Suite\Ultimate Packager\UltimatePackager.exe
EXE Generator	C:\Scripting_Framework\Suite\EXE Generator\EXEGenerator.exe
Script Generator	C:\Scripting_Framework\Suite\Script Generator\Script Generator.exe
SCCM Application Creator	C:\Scripting_Framework\Suite\SCCM Application Creator\SCCM Application Creator.exe
Remote Manager	C:\Scripting_Framework\Suite\Remote Manager\RemoteManager.exe
REG File Converter	C:\Scripting_Framework\Suite\REG File Converter\REG File Converter.exe
Manual English	C:\Scripting_Framework\Suite\Manuals\English.pdf
Manual German	C:\Scripting_Framework\Suite\Manuals\German.pdf
Help Tools	C:\Scripting_Framework\Suite\Help Tools\HelpTools.exe
Support Collector Tool	C:\Scripting_Framework\Suite\Support Collector Tool\Support Collector Tool.exe

Instructions for each tool can be found in the manual.



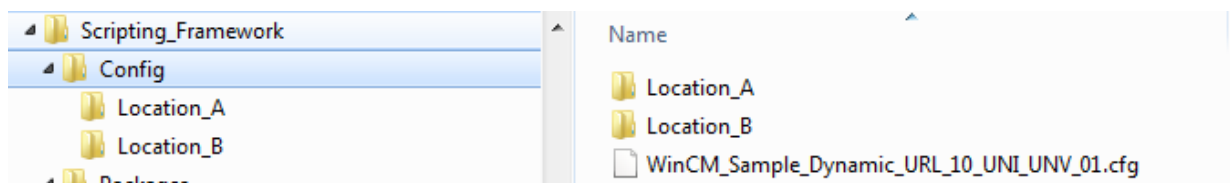
1.2.3 Workplace Folder Structure

The *Scripting_Framework* folder is located directly on the C: drive. It contains everything needed to create and test the packages.

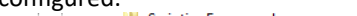
	<p>Config -> Local storage for config files in which configurations are stored. Details below.</p> <p>Packages -> Local storage for packages, with two subdirectories for special example packages</p> <p>Suite -> Collection of all modules and manuals</p> <p>Template -> Package template with examples</p>
--	--

1.2.3.1 Configuration Files (C:\Scripting_Framework\Config)

The configuration files for the packages (if required) are located in the **Config** folder. It is divided into two subfolders that contain location-specific settings. For each location, it is therefore possible to define different configurations such as server names, etc. for the same package.



You can find the configuration required for each client in the registry. For this Workplace, one has already been configured.

	ScriptingFramework	c:\Autoupdate\atn	REG_SZ	C:\Scripting_Framework\Autoupdate
	Config	c_configcompany	REG_SZ	Location_A
	Inventory	c_configpath	REG_SZ	C:\Scripting_Framework\Config
	Reboot	Pgm_Don_Ho_Notepad++_691_MUI_UNV_01	REG_SZ	Present
	Variables	PgmDonHoNotepad++	REG_SZ	Present;6.9.1
				REG_SZ
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ScriptingFramework\Variables				

This configuration can be created using the Help Tools. Details can be found in Chapter 2 of the Manual, *Deployment and Configuration*. An explanation of the various package configurations and locations can be found in the Chapter *Loading Package Variables for Dynamic Packages* of the Manual.



Example - Citrix Receiver (Package *Citrix_Receiver_14110012_MUI_UNV_01*)

The log file shows that the config file was first successfully searched for in the folder

C:\Scripting_Framework\Config\Location_A

```
19:10:2016-14:14:26 INFO: ===== LoadVariables =====
19:10:2016-14:14:26 INFO: Folder Level 1: C:\Scripting_Framework\Config\Location_A\Citrix_Receiver_14110012_MUI_UNV_01.cfg
19:10:2016-14:14:26 INFO: INISection: CommonClientSettings
19:10:2016-14:14:26 INFO: Store0 = Location_A;https://LocationA.local/citrix/winCM/discovery
19:10:2016-14:14:26 INFO: ===== Taskkill =====
```

Example - WinCM Sample Dynamic URL (Package *WinCM_Sample_Dynamic_URL_10_UNI_UNV_01*)

The log file shows that the config file was first unsuccessfully searched for in the folder

C:\Scripting_Framework\Config\Location_A, but was then successfully searched for in the folder

C:\Scripting_Framework\Config in the second step.

```
19:10:2016-14:38:07 INFO: ===== LoadVariables =====
19:10:2016-14:38:07 INFO: Folder Level 1: C:\Scripting_Framework\Config\Location_A\WinCM_Sample_Dynamic_URL_10_UNI_UNV_01.cfg
19:10:2016-14:38:07 INFO: Folder Level 2: C:\Scripting_Framework\Config\WinCM_Sample_Dynamic_URL_10_UNI_UNV_01.cfg
19:10:2016-14:38:07 INFO: INISection: CommonClientSettings
19:10:2016-14:38:07 INFO: ShortcutTarget = http://www.scriptingframework.ch
19:10:2016-14:38:07 INFO: ===== Taskkill =====
```

This example shows how the Powershell script *Launcher.ps1* is started first when starting a software (scripting mode). Before Internet Explorer is started, the configuration file in which the URL is stored is read. Internet Explorer is then started with this URL.

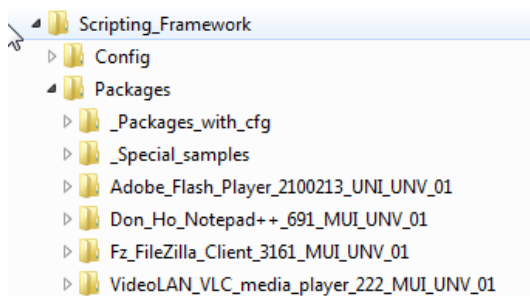
```
1 # =====
2 # Scripting Mode
3 # =====
4
5 # Load Variables (NoErrors for offline clients - If offline, the last loaded are used)
6 SF-LoadVariables "%c_ConfigPath%\%_PkgCFGName%.cfg" "CommonClientSettings" -User -NoErrors
7 If ((SF-Variables "%_VarPrefix-ShortcutTarget%") -eq "") {SF-Exit "Value for Variable <ShortcutTarget> must be"
8
9 # Run Application without Wait
10 SF-Run "%_ProgramFiles32%\Internet Explorer\iexplore.exe" "%_VarPrefix-ShortcutTarget%" -Show -NoErrors
11
```

If the URL changes later, only the URL in the configuration file needs to be adjusted. This ensures that a new package does not have to be created and distributed if the URL is changed.

1.2.3.2 Software Packages (*C:\Scripting_Framework\Packages*)

A few example packages are available in the folder *C:\Scripting_Framework\Packages*

You can find additional packages by following the link <http://www.scriptingframework.ch/downloads/>



The two subfolders contain packages with configuration files.

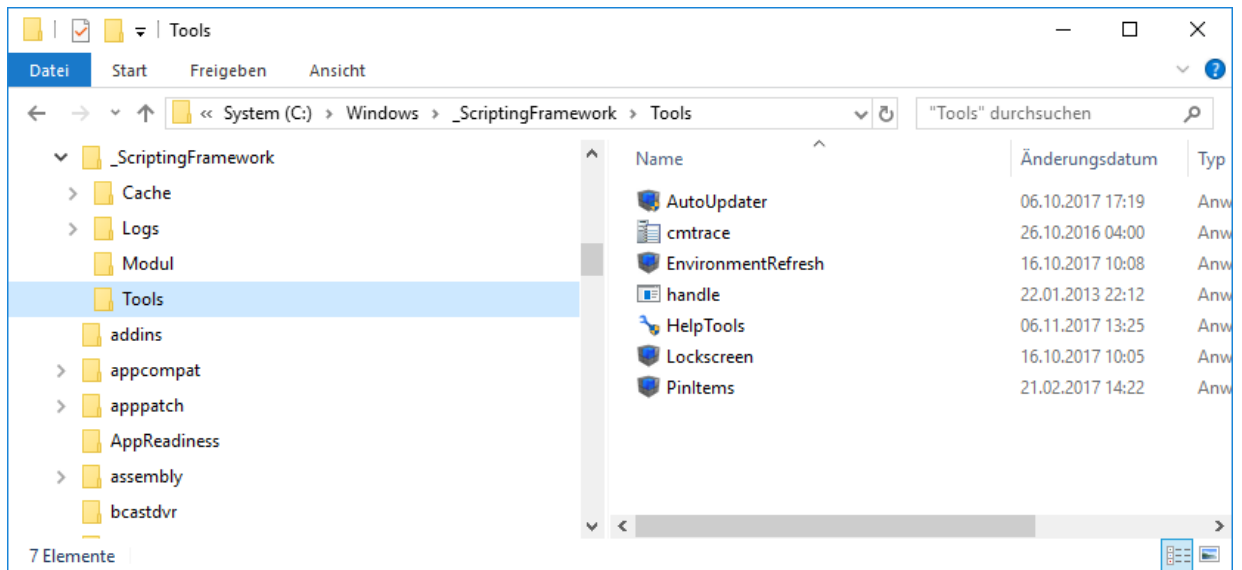
- *C:\Scripting_Framework\Packages_Packages_with_cfg*
- *C:\Scripting_Framework\Packages_Special_samples*

The active configuration files are located in the directory *C:\Scripting_Framework\Config*



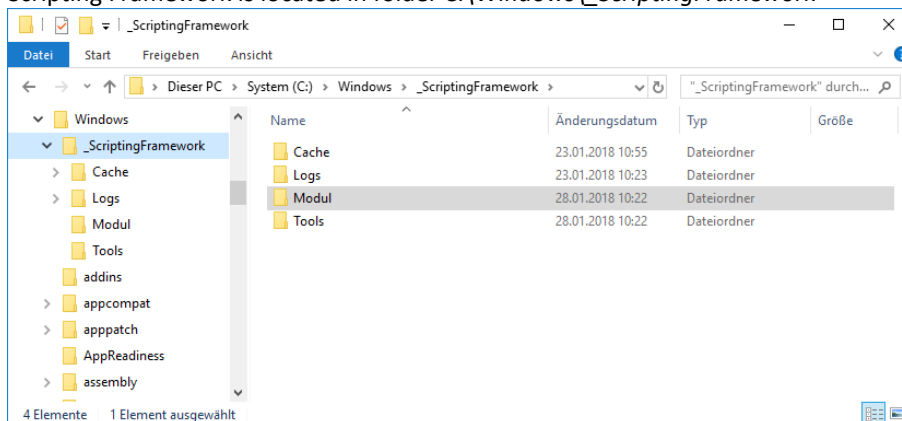
1.2.3.3 Scripting Framework Suite

All *Scripting Framework* tools are available in the folder *C:\Scripting_Framework\Suite*



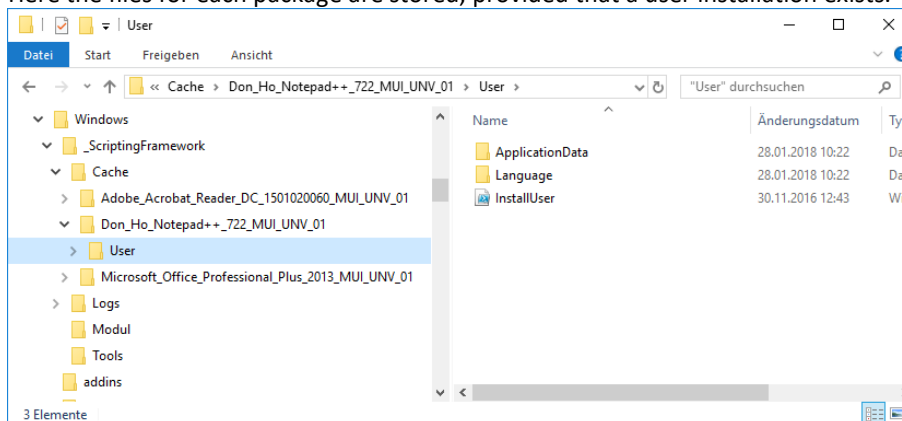
1.2.3.4 Scripting Framework Engine

Scripting Framework is located in folder *C:\Windows_ScriptingFramework*



Cache

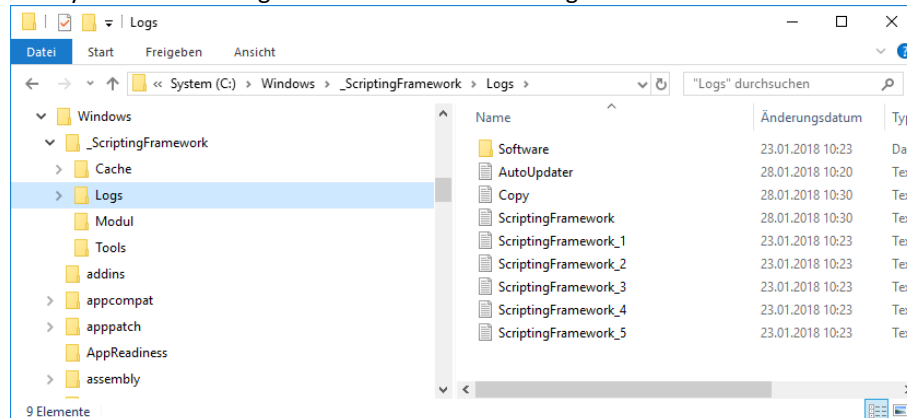
Here the files for each package are stored, provided that a user installation exists.





Logs (Machine)

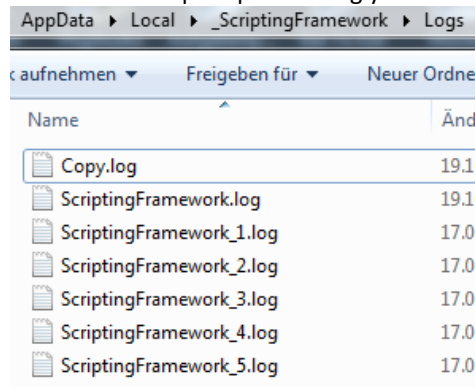
Here you will find all log files that are created during the installation of the software packages.



Once log files reach a certain size, they are split up.

Logs (User)

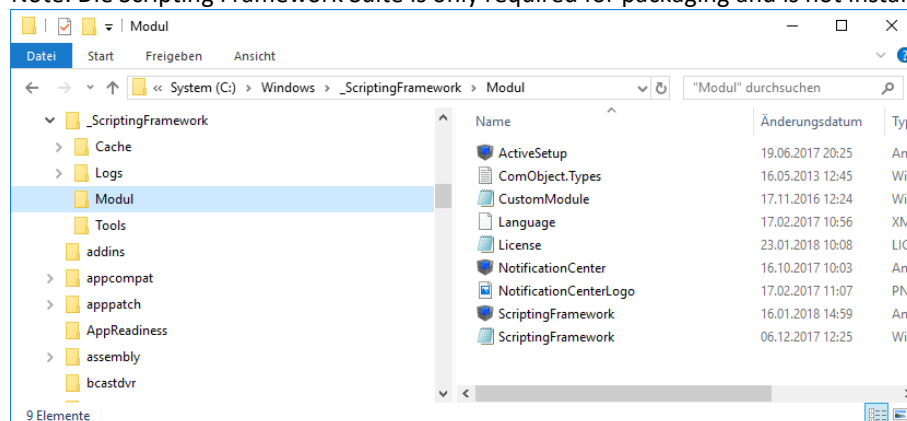
The log files of the user are located under %LocalAppData%_ScriptingFramework\Logs. These are also split up accordingly.



Module

This contains the actual Scripting Framework files that must be present on each client, including the license file.

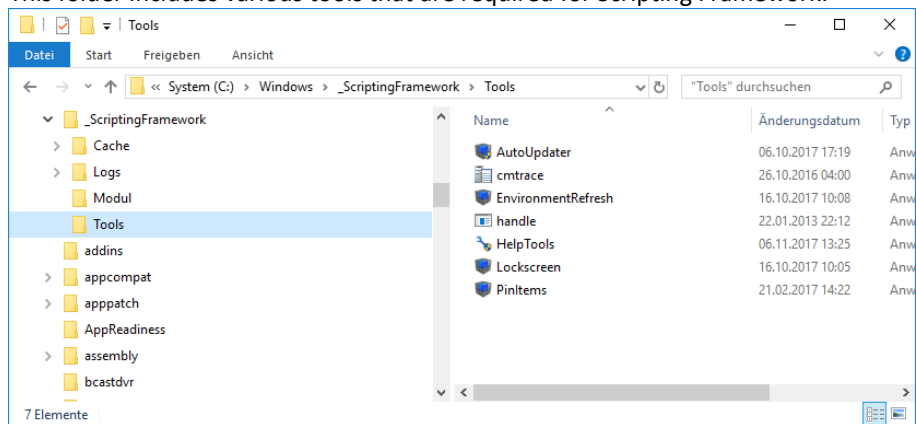
Note: Die Scripting Framework Suite is only required for packaging and is not installed on the clients.





Tools

This folder includes various tools that are required for Scripting Framework.

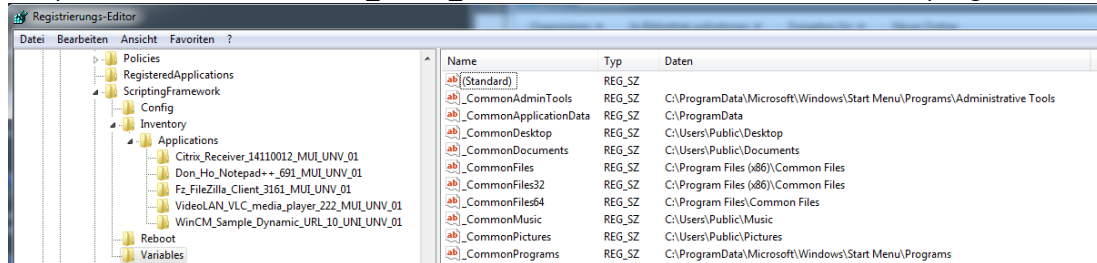




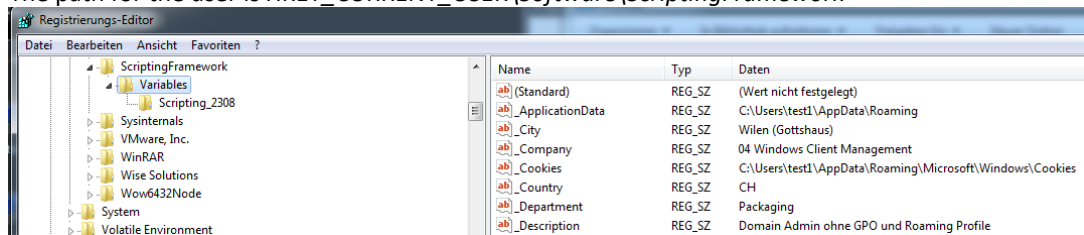
1.2.4 Scripting Framework Registry

In the registry, there is one folder each for the machine part and for the user. These contain all the necessary information.

The path for the machine is `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ScriptingFramework`

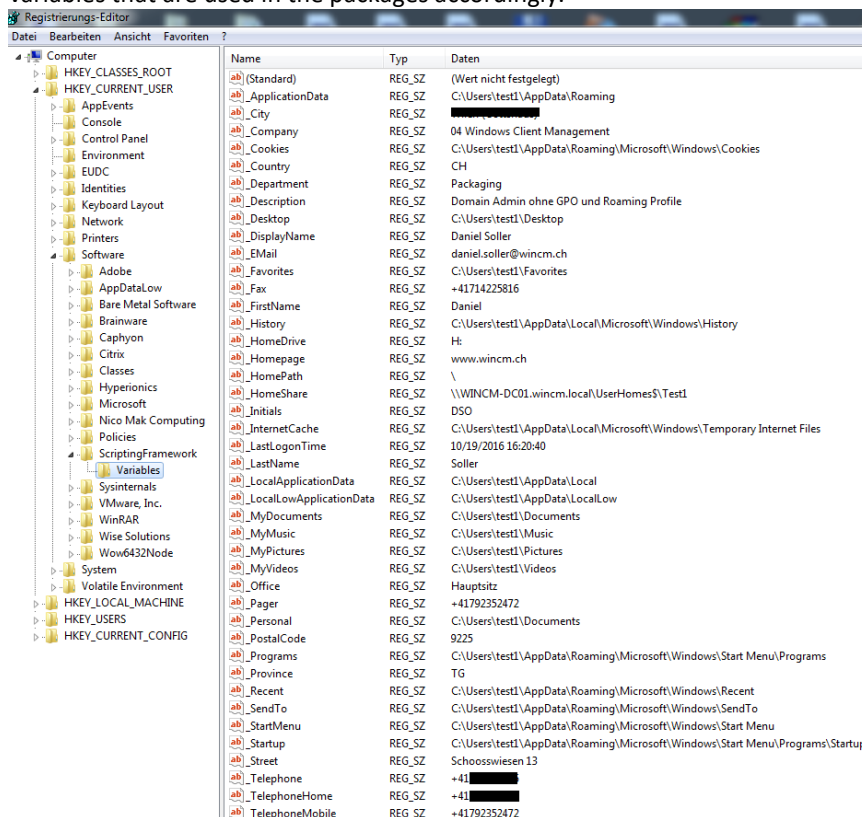


The path for the user is `HKEY_CURRENT_USER\Software\ScriptingFramework`



Tip: Take the time to familiarize yourself with the registry.

If you have logged on as a domain user, you will find information (this is generated dynamically each time) as variables that are used in the packages accordingly.



A detailed description can be found in Chapter 3 of the Manual, *Structure of the Toolkit and Further Information*.

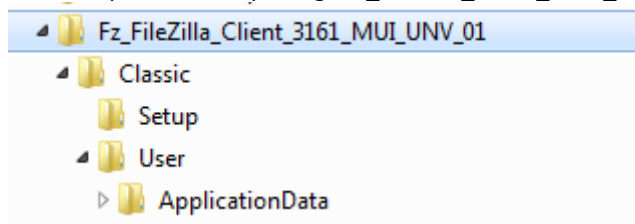


1.2.5 Structure of the Scripting Framework Packages

Each package has a predetermined structure, which is always the same. Refer to the example packages that were installed with the Scripting Framework Workplace. This will help you quickly get an overview of the structure.

A detailed description can be found in Chapter 5 of the Manual, *Scripting Framework Software Package (Definition)*.

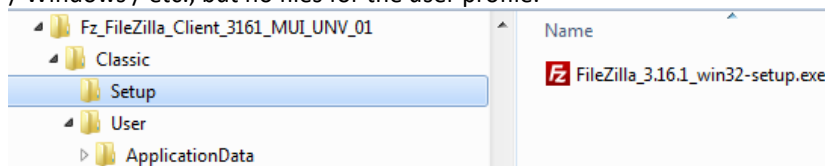
For example, take the package *Fz_FileZilla_Client_3161_MUI_UNV_01*



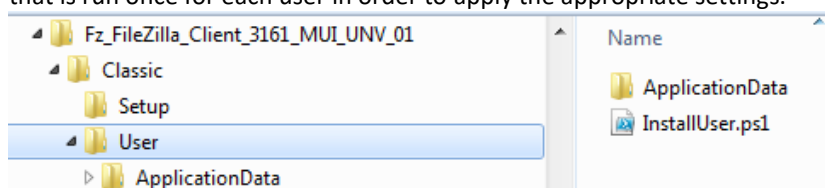
The folder name of a package is based on the following naming convention:
Manufacturer_ProductName_Verison_Language_Company_BuildNumber

The *Classic* folder means that this package is a physical installation. For a virtual App-V package, the name given is *AppV*. The reason for this is that a distinction is made as to what type of package it is. The *Classic* or *AppV* folder is not required.

The *Setup* folder contains installation resources, e.g. an MSI file or files that are copied to the program directory / Windows / etc., but no files for the user profile.

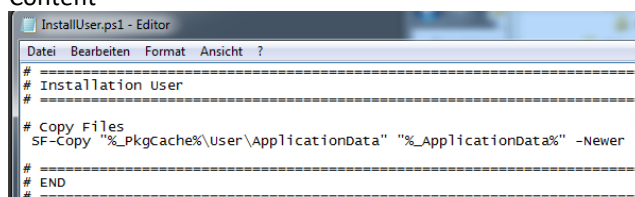


The *User* folder contains all files for the user profile or the user home, for example. It also contains the script that is run once for each user in order to apply the appropriate settings.



In this example package, the user settings are activated – see *InstallUser.ps1*. These are applied once automatically when a user logs on to the client. If the user is already logged on and the package is installed on the client, the settings are applied directly without a renewed login. This also applies to terminal servers for all registered users. However, this function can be deactivated via a configuration in the registry.

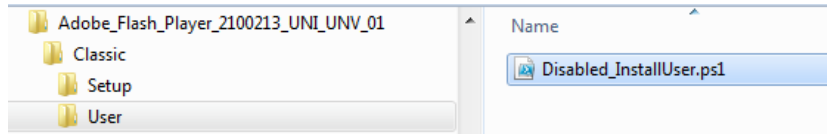
Content





In the next example (Adobe Flash Player), no user part is required. It is therefore an installation that applies exclusively to the client. For this reason, the user part has been deactivated in this package. The .ps1 script in the *User* folder must be renamed to *Disabled_InstallUser.ps1* for this.

It is essential that the file is present if no installation is required for the user. Do not delete it under any circumstances.



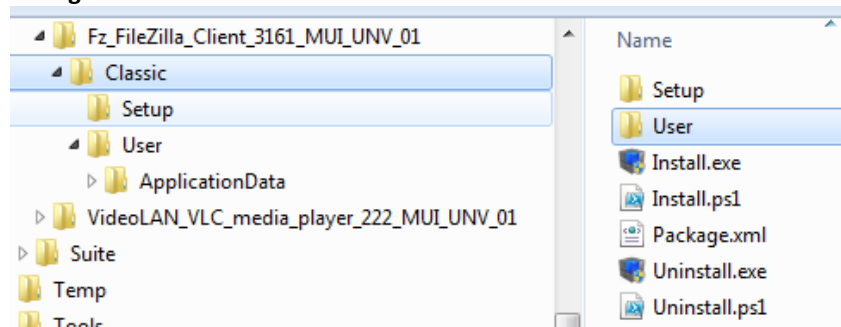
Content of *Disabled_InstallUser.ps1*

```

Disabled_InstallUser.ps1 - Editor
Datei Bearbeiten Format Ansicht ?
# =====
# Installation User
# =====
# No Action
# =====
# END
# =====

```

Configuration and Installation Files



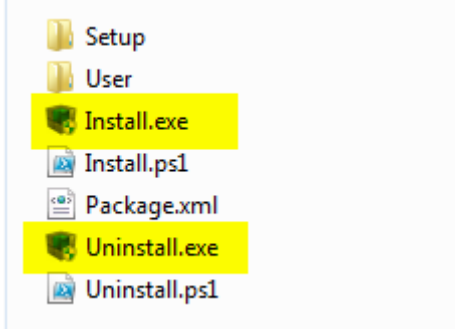
Install.exe	This is used to install files
Install.ps1	The install script
Package.xml	The definition for the package
Uninstall.exe	This is used to uninstall files
Uninstall.ps1	The uninstall script

You can edit the ps1 files (PowerShell) with the *PowerShell ISE Editor* and run them directly. Of course, other editors with which the script cannot be run directly can also be used.



1.2.6 Installing and Uninstalling Packages Manually

The Install.exe file is run to install packages and the Uninstall.exe file to uninstall them.



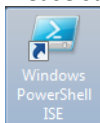
If a user part is defined in the package, it is run automatically and made visible by means of a status window.

1.2.7 Creating Packages

Use the *Ultimate Packager* to create packages if a silent installation is not possible or certain settings are required after the installation of a software.

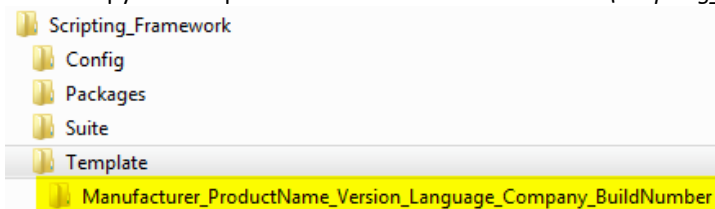
As mentioned above, we will soon offer the *Script Generator*. Until the former is ready, the PowerShell Editor can assist you.

Please start the desktop shortcut *Windows PowerShell ISE* with administrator rights.



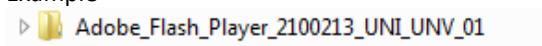
Scripting Framework's PowerShell library is automatically loaded.

Please copy the template that is located in the folder *C:\Scripting_Framework\Template*



Rename the template accordingly.

Example



Next, fill out the Package.xml accordingly.

```
<!-- Package Definition -->
<Package>
  <BuildNumber>01</BuildNumber>
  <Company>UNV</Company>
  <Manufacturer>Adobe</Manufacturer>
  <ProductName>Flash Player</ProductName>
  <Version>21.0.0.213</Version>
  <VersionShort>20</VersionShort>
  <Language>UNI</Language>
</Package>
```



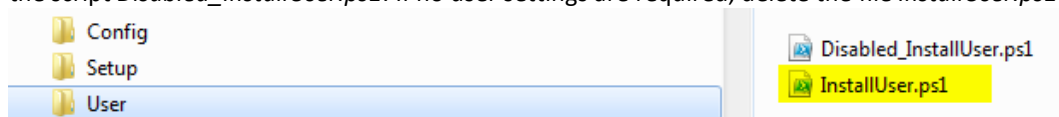
Now create the script with the *Scripting Framework Functions*. The template contains frequently used function examples and mechanisms.

```

Install.ps1 X
22 # =====
23 # Some Examples - Delete if they are not needed
24 # =====
25
26 # Check OStype
27 If ((SF-Variables "%_OStype%") -eq "x64") {SF-Exit "This Software is only for Windows 32bit"}
28 If ((SF-Variables "%_OStype%") -eq "x86") {SF-Exit "This Software is only for Windows 64bit"}
29
30 # No install on Workstation and Notebook
31 If ((SF-Variables "%_ComputerType%") -eq "WORKSTATION") {SF-Exit "This Software is not for Workstations. Exit Installation"}
32 If ((SF-Variables "%_ComputerType%") -eq "NOTEBOOK") {SF-Exit "This Software is not for Notebooks. Exit Installation"}
33
34 # Load Variables
35 SF-LoadVariables "%c_ConfigPath%\%_PkgCFGName%.cfg" "CommonClientSettings"
36 If ((SF-Variables "%_VarPrefix-XXX%") -eq "") {SF-Exit "Value for Variable <XXX> must be set in %_PkgCFGName%.cfg"}
37
38 # Terminate Process
39 SF-Taskkill "Prima5.exe"
40
41 # Terminate all Processes and Handles in a Folder
42 SF-Taskkill "%_SystemDrive%\TestFolder"
43
44 # Language (Machine) - Function to create _PkgLang Variable
45 SF-Language "1031,1033,1036,1040" "1033"
46
47 "..."

```

If you require user settings (files, registry, shortcuts, etc.), edit the script *InstallUser.ps1* and consequently delete the script *Disabled_InstallUser.ps1*. If no user settings are required, delete the file *InstallUser.ps1*



In *InstallUser.ps1*, you can find frequently used function examples and mechanisms.

```

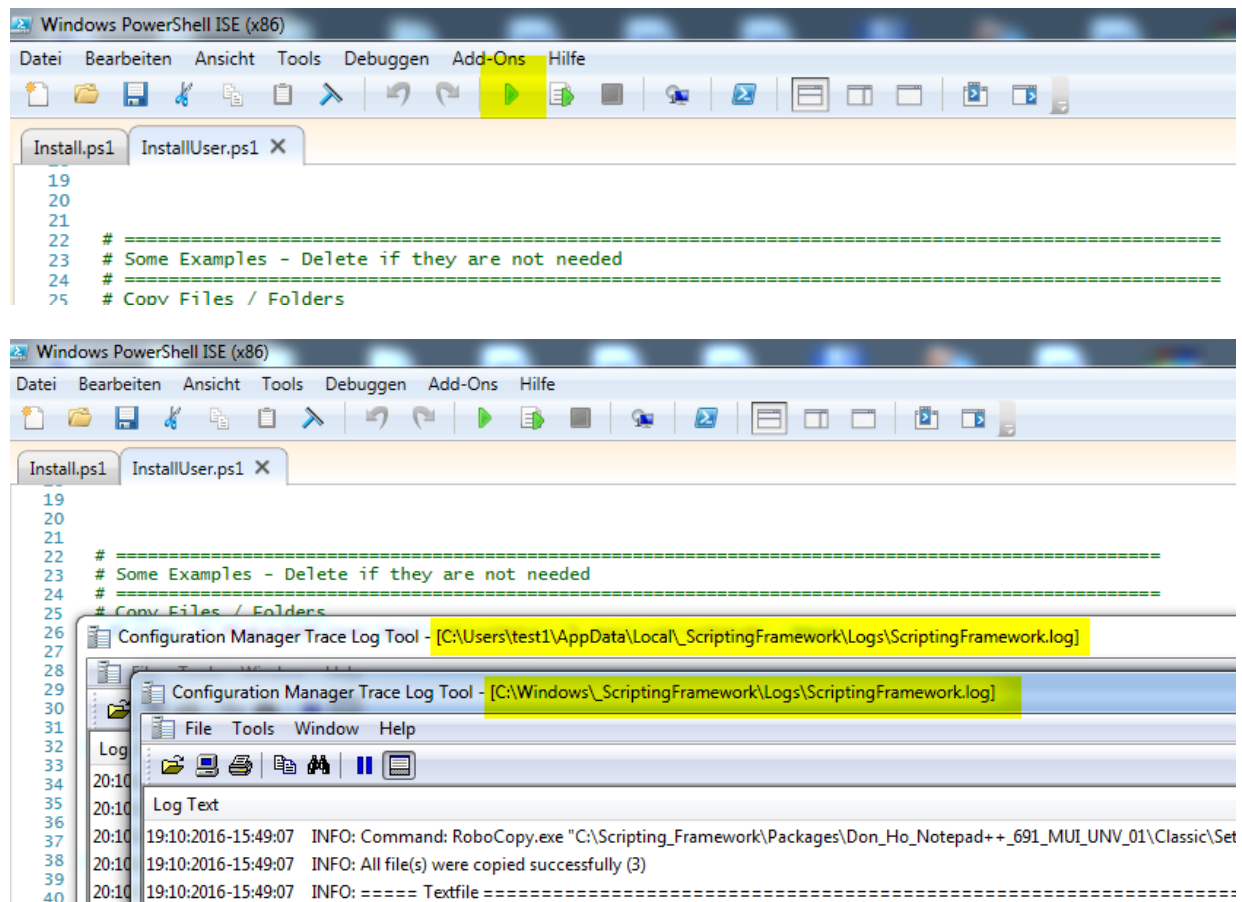
Install.ps1 InstallUser.ps1 X
19
20
21
22 # =====
23 # Some Examples - Delete if they are not needed
24 # =====
25
26 # Copy Files / Folders
27 SF-Copy "%_PkgCache%\User\ApplicationData" "%_ApplicationData%" -Newer
28
29 # Language (User) - Function to create _PkgLang Variable
30 SF-Language "1031,1033,1036,1040" "1033"
31
32 # DEU
33 If ((SF-Variables "%_PkgLang%") -eq "1031") {
34   SF-Register32 "HKey_Current_User" "Software\Adobe\Acrobat Reader\11.0\Language\current" "" "locale\de_de\rdlang32.deu" "REG_SZ"
35   SF-INIWrite "%_ApplicationData%\Canneverbe Limited\CDBurnerXP\UserSettings.ini" "General" "AppLanguage" "de-DE"
36 }
37
38 # ENU
39 If ((SF-Variables "%_PkgLang%") -eq "1033") {
40   SF-Register32 "HKey_Current_User" "Software\Adobe\Acrobat Reader\11.0\Language\current" "" "acrord32.dll" "REG_SZ"
41   SF-INIWrite "%_ApplicationData%\Canneverbe Limited\CDBurnerXP\UserSettings.ini" "General" "AppLanguage" "en-US"
42 }
43
44 # FRA
45 If ((SF-Variables "%_PkgLang%") -eq "1036") {
46   SF-Register32 "HKey_Current_User" "Software\Adobe\Acrobat Reader\11.0\Language\current" "" "locale\fr_fr\rdlang32.fra" "REG_SZ"
47   SF-INIWrite "%_ApplicationData%\Canneverbe Limited\CDBurnerXP\UserSettings.ini" "General" "AppLanguage" "fr-FR"
48 }
49
50 # ITA
51 If ((SF-Variables "%_PkgLang%") -eq "1040") {
52   SF-Register32 "HKey_Current_User" "Software\Adobe\Acrobat Reader\11.0\Language\current" "" "locale\it_it\rdlang32.ita" "REG_SZ"
53   SF-INIWrite "%_ApplicationData%\Canneverbe Limited\CDBurnerXP\UserSettings.ini" "General" "AppLanguage" "it-IT"
54 }
55
56 # Disable Autoupdate
57 SF-INIWrite "%_ApplicationData%\Canneverbe Limited\CDBurnerXP\UserSettings.ini" "General" "AutoUpdate" "0"
58
59 # =====
60 # END
61 # =====

```

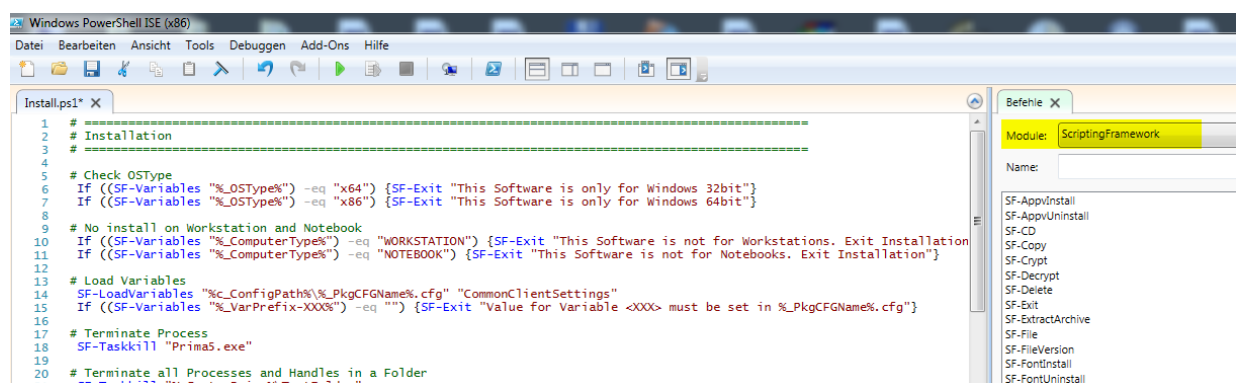
If you run a script in the PowerShell Editor, the Trace Log Tool, which allows you to follow the installation in real time based on the log file, starts automatically.



Two trace windows are opened: one for the software installation and one for the user settings. You can observe all the actions performed except the writing of registry keys.

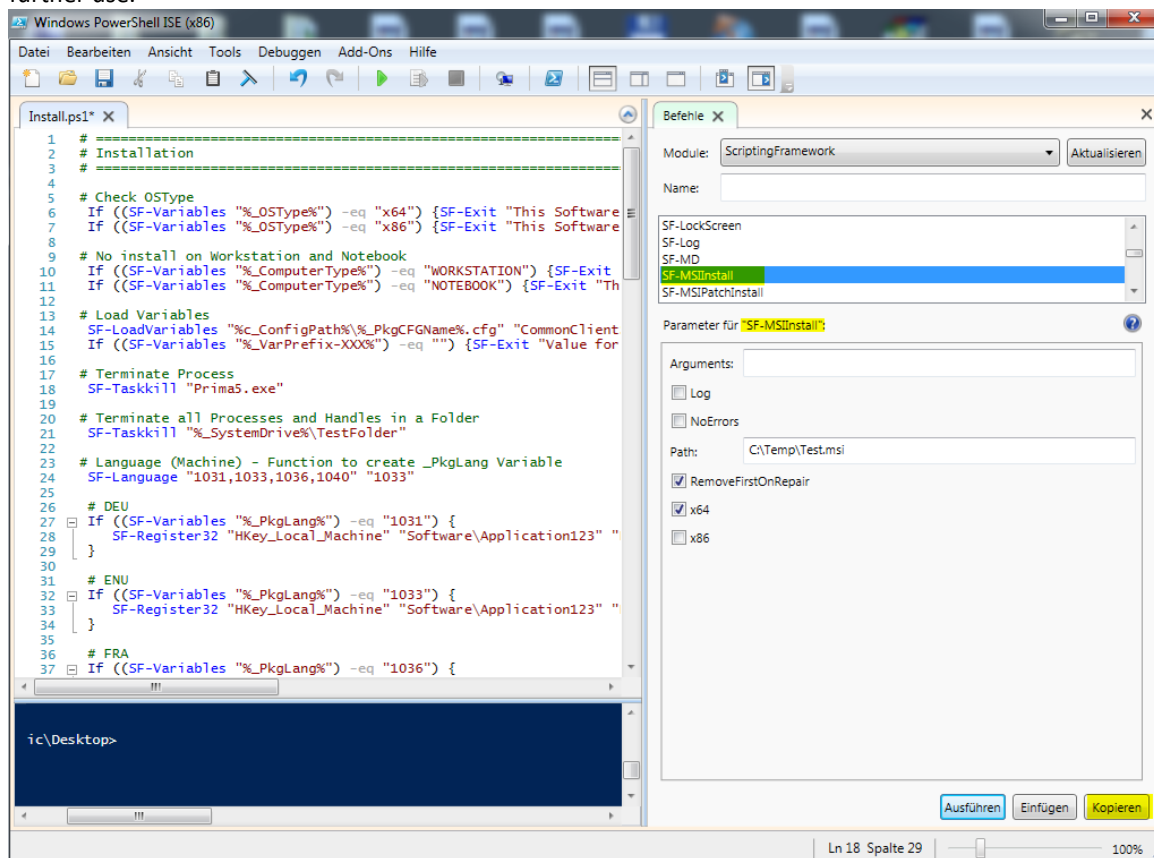


You can view the Scripting Framework functions in the Powershell Editor. To do this, select the ScriptingFramework module.





After you click on the function, the available parameters are displayed in the lower part of the window. Complete the parameterization. Then use the [Copy] button to copy the command line to the clipboard for further use.



SF-MSIInstall -Path C:\Temp\Test.msi -RemoveFirstOnRepair -x64

1.2.8 Scripting Framework Functions

There are currently more than 60 functions available:

- Installing and uninstalling of applications
- Deleting of files and folders, including the automatic closure of active executables
- Writing of registry values (32-bit and 64-bit)
- Copying of files
- Installing and uninstalling of AppV packages
- Creating shortcuts
- Installing fonts
- Creating services
- etc.

Of course, your own PowerShell code can also be used within the Scripting Framework packages.



1.3 Closing Remarks

If you have any questions or comments about Scripting Framework, please do not hesitate to contact us.

E-mail info@wincm.ch

We wish you much success and fun with *Scripting Framework*.