

Scripting Framework PowerShell Toolkit

Installation eines Workplace für die Paketierung und Test's





1	Insta	llation des Scripting Framework Workplace	
1.1		ation durchführenation durchführen	
1.2	Einführung in Scripting Framework		
1.2	Ellilui	irung in Scripting Framework	
	1.2.1	Das Startmenu und die Desktop Verknüpfungen	
	1.2.2	Scripting Framework Suite	8
	1.2.3	Die Ordnerstruktur des Workplace	
	1.2.3.1	Konfigurations-Files (C:\Scripting_Framework\Config)	
	1.2.3.2	Software Pakete (C:\Scripting_Framework\Packages)	10
	1.2.3.3	Scripting Framework Suite	12
	1.2.3.4	Die Scripting Framework Engine	12
	1.2.4	Die Scripting Framework Registry	
	1.2.5	Aufbau der Scripting Framework Pakete	10
	1.2.6	Manuelle Installation und De-Installation der Pakete	19
	1.2.7	Pakete erstellen	19
	1.2.8	Scripting Framework Funktionen	
1.3	Schlus	sswort	2:



1 Installation des Scripting Framework Workplace

Diese Kurzanleitung verschafft Ihnen einen optimalen Überblick von Scripting Framework. Bitte lesen Sie das Handbuch für detailliertere Informationen.

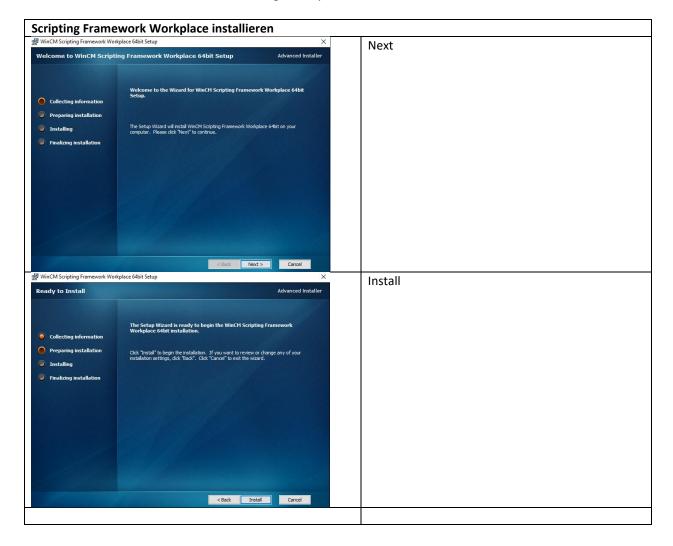
Installieren Sie auf Ihrem Packaging-Client schnell und einfach einen kompletten Workplace, für das Paketieren und Testen von Software-Paketen. Netzwerk-Shares sind keine erforderlich, sowie auch kein Software-Verteilsystem.

Anforderungen

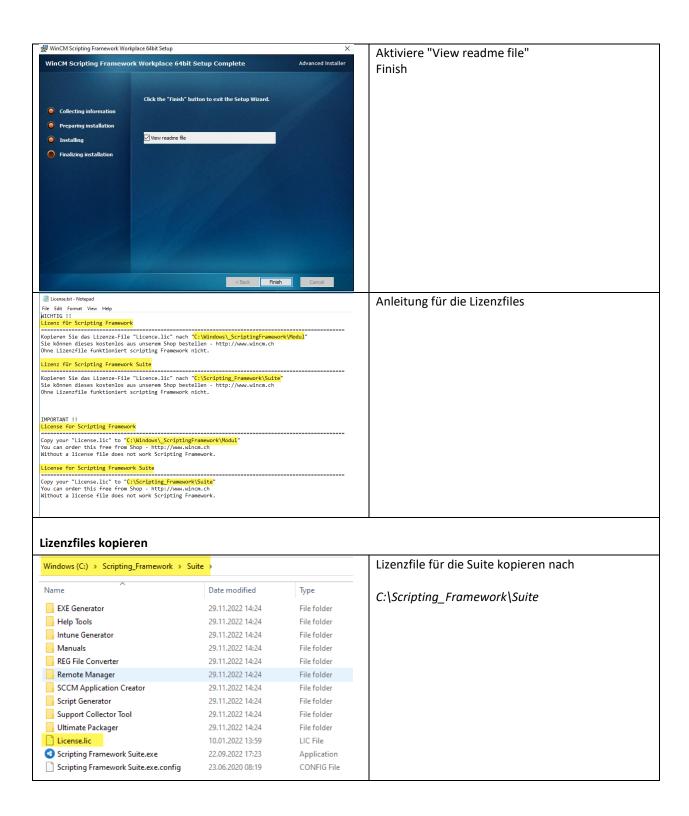
- Windows 64bit Betriebssystem
- Microsoft .Net Framework ab Version 4.5
- PowerShell ab Version 4.0
- Lizenzfiles für Engine und Suite Auf unserer Website können Sie eine kostenlose Trial Lizenz bestellen

1.1 Installation durchführen

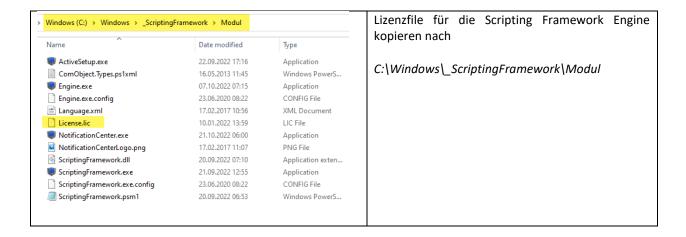
Führen sie die Datei WinCM_Scripting_Framework_Workplace_x64.msi als Administrator aus. Installiert wird Framework mit Suite und einigen Beispiel-Paketen.









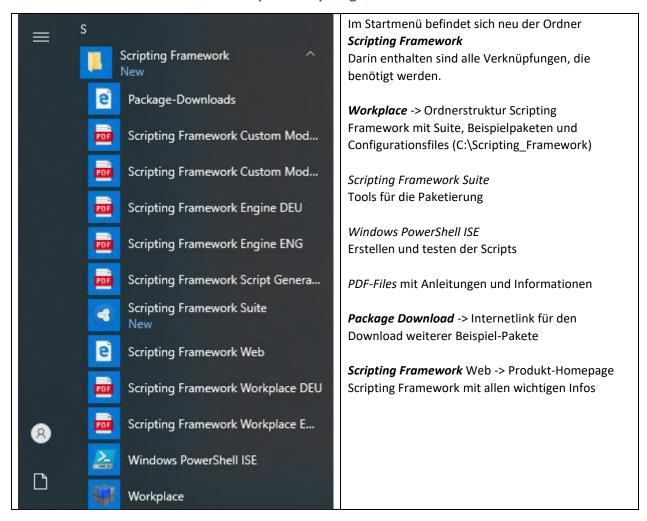




1.2 Einführung in Scripting Framework

Damit Sie sich im *Scripting Framework Workplace* schnell zurechtfinden, können Sie sich nun einen Überblick verschaffen.

1.2.1 Das Startmenu und die Desktop Verknüpfungen



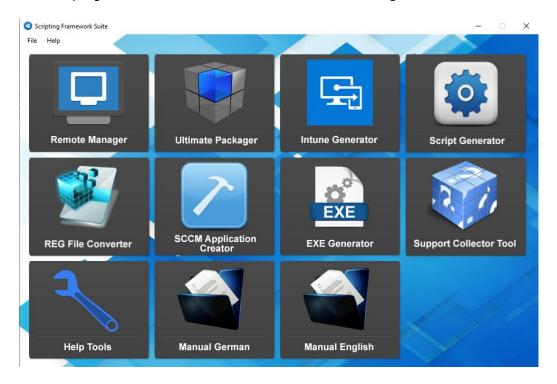


Die wichtigsten Verknüpfungen sind auf dem Desktop vorhanden. Scripting Framework Suite Scripting Nützliche Tools für die Paketierung Framewo... Windows PowerShell ISE Erstellen und testen der Scripts Workplace Öffnet *C:\Scripting_Framework* PowerSh... Darin befindet sich das komplette Framework (ohne Engine) Trace-Copy **Logfile Copy** Trace-Machine Logfile Install.ps1 Trace-User Trace-Copy Logfile InstallUser.ps1



1.2.2 Scripting Framework Suite

In der Scripting Framework Suite sind verschiedene Tools zusammengefasst.

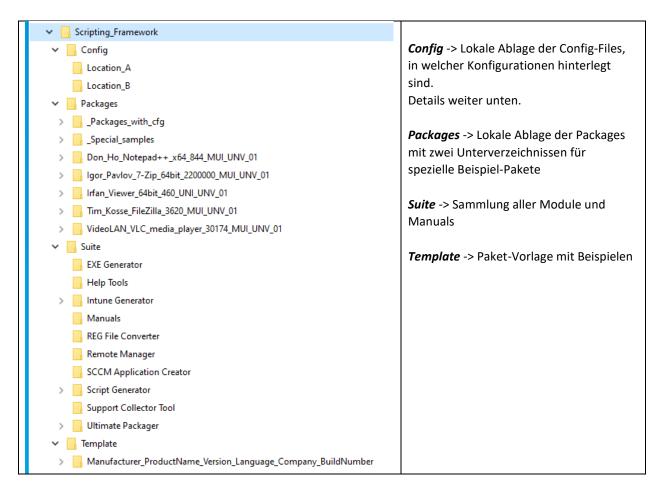


Eine Anleitung der einzelnen Tools finden Sie im Manual Scripting Framework Engine DEU.pdf



1.2.3 Die Ordnerstruktur des Workplace

Direkt im Laufwerk C: befindet sich der Ordner *Scripting_Framework*. Darin ist alles enthalten, um die Packages zu erstellen und zu testen.

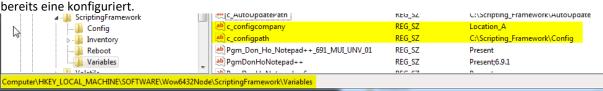


1.2.3.1 Konfigurations-Files (C:\Scripting_Framework\Config)

Im Ordner *Config* befinden sich die Konfigurationsfiles der Pakete (wenn benötigt). Er ist unterteilt in zwei Unterordner, welche die Standort-Spezifischen Einstellungen enthalten. Somit ist es möglich, pro Standort verschiedene Konfigurationen wie Servernamen usw. für dasselbe Paket zu definieren.



Die Konfiguration welche bei jedem Client erforderlich ist, finden Sie in der Registry. Für diesen Workplace ist





Mit dem Help Tools kann diese Konfiguration erzeugt werden. Details im Handbuch Kapitel 2 *Bereitstellung und Konfiguration*. Die Erklärung wie es mit den verschiedenen Konfigurationen der Pakete sowie der Standorte finden Sie im Handbuch Kapitel Laden von *Paketvariablen für dynamische Pakete*.

Beispiel - Citrix Receiver (Paket Citrix_Receiver_14110012_MUI_UNV_01)

Im Logfile ist ersichtlich, dass als erstes im Ordner C:\Scripting_Framework\Config\Location_A nach dem Config-File erfolgreich gesucht wurde.

Beispiel - WinCM Sample Dynamic URL (Paket WinCM_Sample_Dynamic_URL_10_UNI_UNV_01)

Im Logfile ist ersichtlich, dass als erstes im Ordner C:\Scripting_Framework\Config\Location_A\ nach dem Config-File erfolglos gesucht wurde, aber dann im zweiten Schritt erfolgreich im Ordner C:\Scripting_Framework\Config

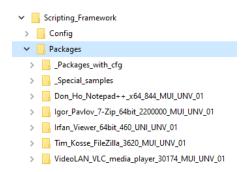
Dieses Beispiel zeigt auf, wie beim Start einer Software zuerst ein Powershell Script *Launcher.ps1* gestartet wird (Scripting Mode). Vor dem Start des Internet Explorer wird das Konfigurationsfile ausgelesen, in welchem die URL hinterlegt ist. Dann wird der Internet Explorer mit dieser URL gestartet.

Wechselt die URL zu einem späteren Zeitpunkt, muss nur die URL im Konfigurationsfile angepasst werden. Damit wird erreicht, dass bei einem Wechsel der URL nicht ein neues Paket erstellt und verteilt werden muss.

1.2.3.2 Software Pakete (C:\Scripting_Framework\Packages)

Im Ordner *C:\Scripting_Framework\Packages* finden Sie ein paar Beispielpakete.

Weitere Pakete stehen Ihnen unter dem folgenden Link http://www.scriptingframework.ch/downloads/ zur Verfügung.





In den beiden Unterordnern befinden sich Pakete mit Konfigurations-Files.

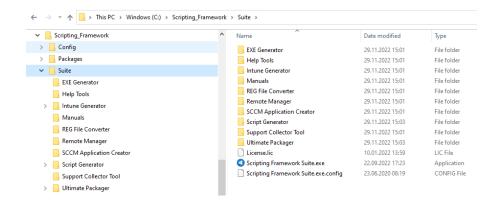
- C:\Scripting_Framework\Packages_Packages_with_cfg
- C:\Scripting_Framework\Packages_Special_samples

Die aktiven Konfigurationsfiles befinden sich im Verzeichnis C:\Scripting_Framework\Config



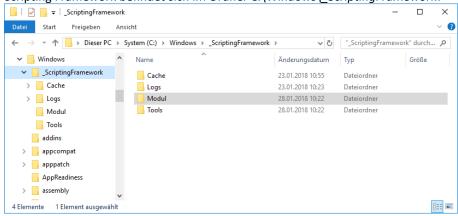
1.2.3.3 Scripting Framework Suite

Im Ordner C:\Scripting_Framework\Suite sind alle Tools von Scripting Framework verfügbar.



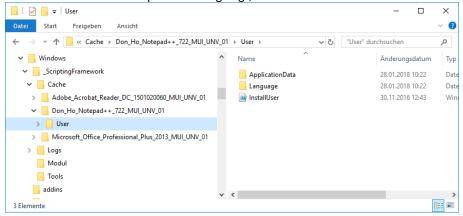
1.2.3.4 Die Scripting Framework Engine

Scripting Framework befindet sich im Ordner C:\Windows\ ScriptingFramework.



Cache

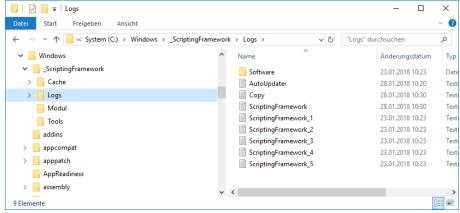
Darin werden die Dateien pro Paket abgelegt, sofern eine User-Installation vorhanden ist.





Logs (Machine)

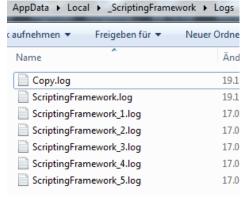
Hier finden Sie alle Logdateien, welche mit der Installation der Software Pakete erzeugt werden.



Ab einer bestimmten Grösse der Logfiles, werden diese aufgesplittert.

Logs (User)

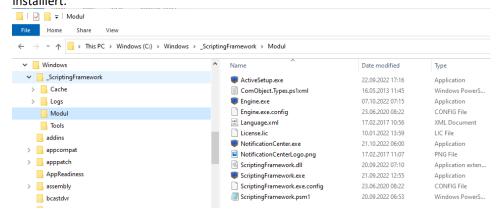
Die Logfiles des Users befinden sich unter $\LocalAppData\%\LocalAppData\%\LocalAppData\%$ Auch diese werden entsprechend aufgesplittert.



Modul

Darin sind die eigentlichen Scripting Framework Dateien enthalten, die auf jedem Client vorhanden sein müssen, inklusive dem Lizenzfile.

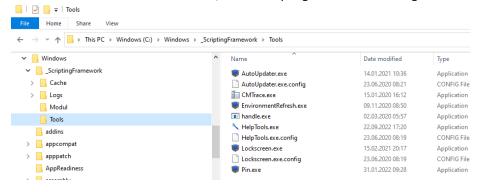
Hinweis: Die Scripting Framework Suite wird nur für die Paketierung benötigt und wird nicht auf die Clients installiert.





Tools

Hier befinden sich verschiedene Tools, die für Scripting Framework benötigt werden.

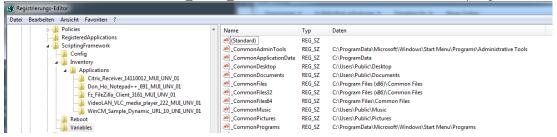




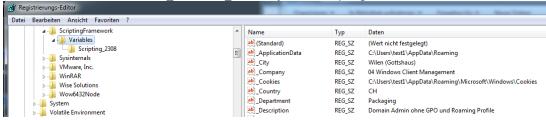
1.2.4 Die Scripting Framework Registry

Es gibt in der Registry je einen Order für den Maschinen Teil und einen für den User. Darin sind alle notwendigen Informationen abgelegt.

Der Pfad für die Machine ist HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ScriptingFramework

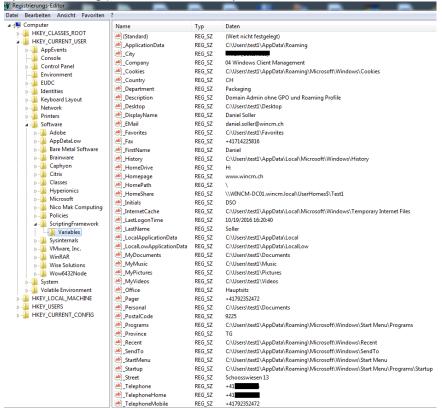


Der Pfad für den User ist HKEY_CURRENT_USER\Software\ScriptingFramework



Tipp - Nehmen Sie sich die Zeit, die Registry genau anzuschauen.

Wenn Sie sich mit einem Domain User angemeldet haben, finden Sie Informationen (diese werden jedes Mal dynamisch erzeugt) als Variablen, die entsprechend in den Paketen verwendet werden.



Die genaue Beschreibung finden Sie im Handbuch Kapitel 3 Aufbau des Toolkits und weitere Informationen

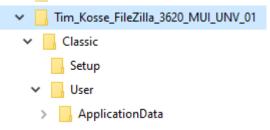


1.2.5 Aufbau der Scripting Framework Pakete

Jedes Paket hat eine vorgegebene Struktur, die immer gleich ist. Orientieren Sie sich an den Beispiel-Paketen, die mit dem Scripting Framework Workplace installiert worden sind. So erhalten Sie sehr schnell eine Übersicht über den Aufbau.

Die ausführliche Beschreibung finden Sie im Handbuch Kapitel 5 – Scripting Framework Software Paket (Definition).

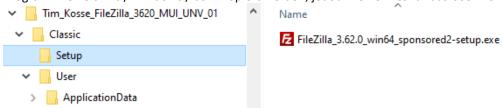
Nehmen wir als Beispiel das Paket Tim_Kosse_FileZilla_3620_MUI_UNV_01



Die Ordnerbezeichnung eines Paketes hat folgende Namenskonvention: Manufacturer_ProductName_Verison_Language_Company_BuildNumber

Der Ordner *Classic* bedeutet, dass es sich bei dem Paket um eine physische Installation handelt. Bei einem virtuellen App-V Paket ist die Bezeichnung jeweils *AppV*. Der Grund dafür ist, dass unterschieden wird, um was für einen Typ von Paket es sich handelt. Der Ordner *Classic* oder *AppV* ist nicht erforderlich.

Im Ordner *Setup* befinden sich Installationssourcen, z.B. eine MSI-Datei, oder Dateien welche ins Programmverzeichnis / Windows / usw. kopiert werden, jedoch keine Files für das UserProfile.



Im Ordner *User* sind alle Dateien für das UserProfile oder z.B. Userhome. Darin befindet sich auch der Script, welcher für jeden User einmal ausgeführt wird, um die entsprechenden Einstellungen vorzunehmen.



In diesem Beispiel Paket sind die Benutzereinstellungen aktiviert, siehe *InstallUser.ps1*. Diese werden einmalig automatisch ausgeführt, wenn sich ein User an dem Client anmeldet. Falls der User bereits angemeldet ist und das Paket auf den Client installiert wird, werden die Einstellungen direkt ohne neue Anmeldung angewendet. Dies gilt auch für Terminalserver für alle angemeldeten User. Diese Funktion lässt sich jedoch über eine Konfiguration in der Registry deaktivieren.

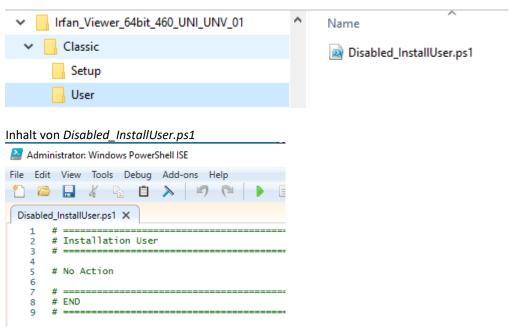


Inhalt

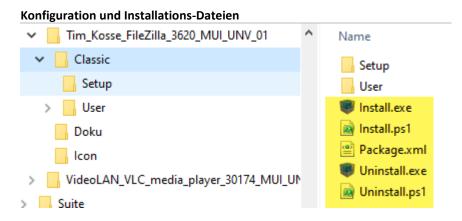
```
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
InstallUser.ps1 X
       # Installation User
       # Copy Files/Folders
       SF-Copy "%_PkgCache%\User\ApplicationData" "%_ApplicationData%" -Newer
       # Language (User) - Function to create _PkgLang Variable
SF-Language "1031,1033,1036,1040" "1033"
  10
  11
  # 550
21 ⊡If ((SF-Variables "%_PkgLang%") -eq "1031") {
13 | SF-Textfile "%_ApplicationData%\FileZilla\filezilla.xml" "#Language#" "de" -Replace
      }
  15
       # ENU
  16
  16 # ENU
17 ⊡If ((SF-Variables "%_PkgLang%") -eq "1033") {
18 | SF-Textfile "%_ApplicationData%\FileZilla\filezilla.xml" "#Language#" "en_US" -Replace
      }
  19
  20
  21
  22 = If ((SF-Variables "%_PkgLang%") -eq "1036") {
23 | SF-Textfile "%_ApplicationData%\FileZilla\filezilla.xml" "#Language#" "fr" -Replace
      1
  24
       # ITA
  26
     }
  29
  30
  31
       # END
  32
       #
```

Im nächsten Beispiel (Irfan Viewer) wird kein Userteil benötigt. Es ist also eine Installation, welche ausschliesslich den Client betrifft. Deshalb wurde in diesem Paket der Userteil deaktiviert. Der .ps1 Script im Ordner *User* muss dafür auf den Namen *Disabled_InstallUser.ps1* (ist im Template enthalten) umbenannt werden.

Die Datei muss zwingend vorhanden sein, wenn keine Installation für den User erforderlich ist. Löschen Sie diese in keinem Fall.







Install.exe	Damit wird die Installation durchgeführt
Install.ps1	Der Installations-Script
Package.xml	Die Definition für das Paket
Uninstall.exe	Damit wird die De-Installation durchgeführt
Uninstall.ps1	Der Deinstallations-Script

Die ps1-Dateien (PowerShell) können Sie mit *PowerShell ISE Editor* bearbeiten und direkt ausführen. Selbstverständlich sind auch andere Editoren verwendbar, bei jenen das direkte Ausführen des Scripts nicht möglich ist.



1.2.6 Manuelle Installation und De-Installation der Pakete

Für die Installation wird die Datei Install.exe ausgeführt und für die Deinstallation die Uninstall.exe



Ist im Paket ein Userteil definiert, wird dieser automatisch ausgeführt und mittels eines Statusfensters sichtbar.

1.2.7 Pakete erstellen

Setzen Sie den *Ultimate Packager* ein, um Pakete zu erstellen, falls keine Silent-Installation möglich ist, oder bestimmte Einstellungen nach einer Installation einer Software notwendig sind.

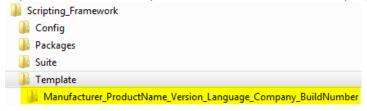
Wie bereits erwähnt, bieten wir demnächst den *Script Generator* an. Bis es soweit ist, unterstützt Sie der PowerShell Editor.

Bitte starten Sie mit Administrations-Rechten die Desktop-Verknüpfung Windows PowerShell ISE.



Die PowerShell-Bibliothek von Scripting Framework wird automatisch geladen.

Kopieren Sie bitte das Template welches sich im Ordner C:\Scripting_Framework\Template befindet.



Benennen Sie das Template entsprechend um.

Beispiel

```
Description  
Adobe_Flash_Player_2100213_UNI_UNV_01
```

Als nächstes füllen Sie das Package.xml entsprechend ab.



Erstellen Sie nun den Script mit den *Scripting Framework Funktionen*. Im Template finden Sie oft verwendete Funktionsbeispiele und Mechanismen.

Benötigen Sie User Einstellungen (Files, Registry, Shortcuts, usw.), bearbeiten Sie den Script *InstallUser.ps1* und löschen demzufolge den Script *Disabled_InstallUser.ps1*. Sind keine Benutzereinstellungen erforderlich, löschen Sie die Datei *InstallUser.ps1*

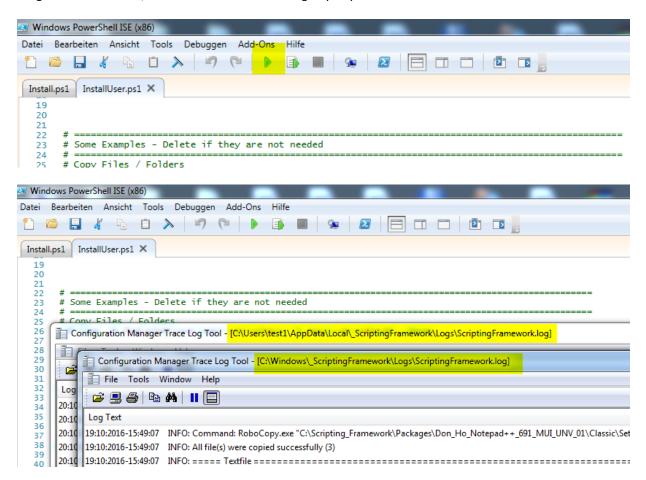


Im InstallUser.ps1 finden Sie oft verwendete Funktionsbeispiele und Mechanismen.

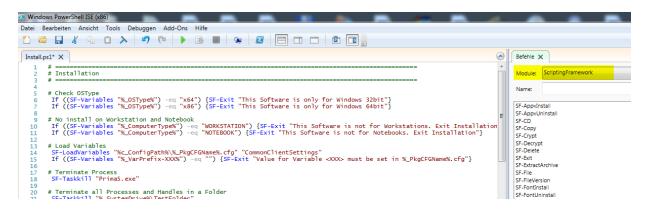
Wenn Sie im PowerShell Editor einen Script ausführen, startet automatisch das Trace Log Tool, mit dem Sie die Installation anhand der Logdatei in Realtime mitverfolgen können. Es werden zwei Trace-



Fenster geöffnet, eines für die Softwareinstallation und das andere für die User-Einstellungen. Sie sehen alle ausgeführten Aktionen, ausser das Schreiben von Registry-Keys.



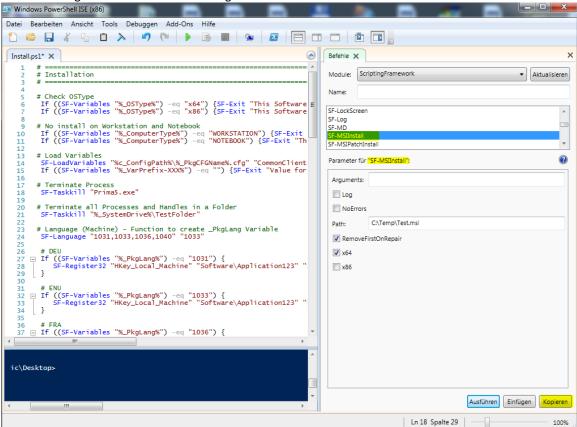
Lassen Sie sich im Powershell Editor die Funktionen von Scripting Framework anzeigen. Dazu wählen Sie das Modul ScriptingFramework aus.





Nachdem Sie auf die Funktion klicken, werden im unteren Bereich die verfügbaren Parameter angezeigt. Nehme Sie die Parametrisierung vor. Danach kopieren Sie mit dem Button [Kopieren] die Befehlszeile in die

Zwischenablage zur weiteren Verwendung.



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

1.2.8 Scripting Framework Funktionen

Es stehen Ihnen aktuell über 70 Funktionen zur Verfügung:

- Installation und Deinstallation von Anwendungen
- Löschen von Dateien und Ordnern inklusiv das automatische Schliessen der aktiven Executables offenen Handels
- Schreiben von Registry Werten (32-Bit und 64-Bit)
- Kopieren von Dateien
- Installation und De-Installation von AppV-Paketen
- Shortcuts erstellen
- Fonts installieren
- Services erstellen
- etc.

Selbstverständlich ist Ihr eigener PowerShell Code innerhalb der Scripting Framework Pakete möglich.



1.3 Schlusswort

Bei Fragen oder Anregungen zu Scripting Framework, zögern Sie bitte nicht uns zu kontaktieren. E-Mail <u>info@wincm.ch</u>

Wir wünschen Ihnen viel Erfolg und Spass mit Scripting Framework.