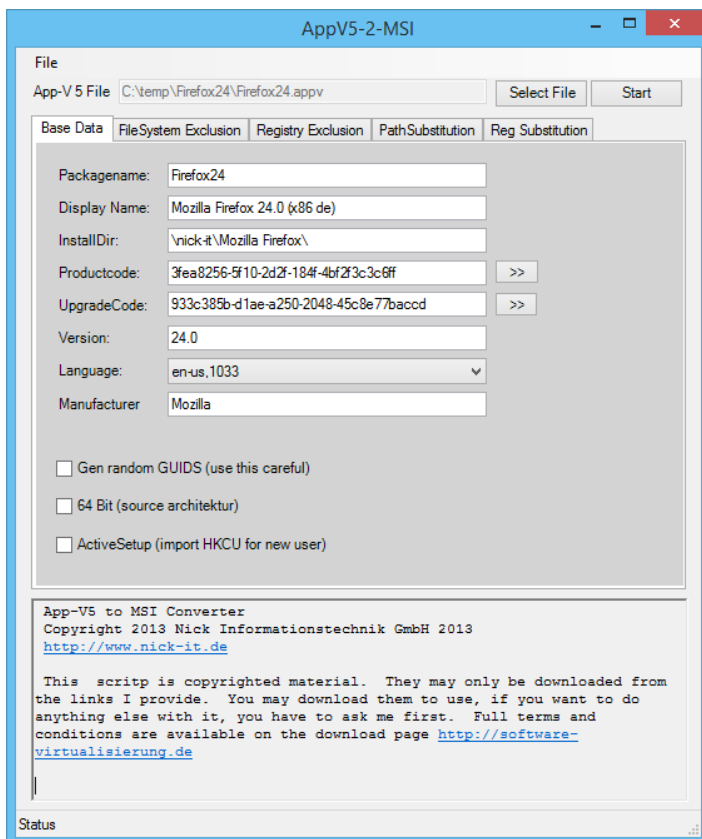


Appv2msi von Nick Informationstechnik

AppV2Msi ein Tool, das ist entstanden ist, um eine .appv Datei (App-V 5) in ein „echtes“ MSI Files zu wandeln. Für mich war das ein Projekt, um tiefer in die Mechanismen der Software-Virtualisierung mit App-V einzusteigen und die Möglichkeiten besser zu verstehen.

Die Entwicklung ist nicht abgeschlossen. Gerade wenn wir entsprechendes Feedback zu der Anwendung erhalten, werden wir diese Anwendung auch weiter entwickeln. Irgendwann vielleicht auch zu einer Pro Version. Aktuell konnten wir mit dem Tool mehr Anwendungen wandeln, als zunächst erwartet.



Zielgruppe

Gerade wurde ich von einem im App-V Bereich darauf angesprochen, warum wir das überhaupt machen. Die einfache Antwort ist, dass es sehr einfach funktioniert!

Folgende Anwendungsszenarien:

Erstellung eines MSI für ein neues „einfaches“ Produkt für die Firmeninterne Nutzung. Man nehme des Sequencer (Technet oder die Beta über connect.microsoft.com – suche nach „desktop optimization Pack“). Und erstellt aus der Sequence ein msi.

Der Weg zurück: Man möchte die Plattform wechseln. Oder eine Anwendung funktioniert nicht als App-V Anwendung. Auch dafür kann das Tool genutzt werden

Inhalt

Appv2msi von Nick Informationstechnik	1
Zielgruppe.....	1
AppV2Msi – Rechtliches (Warnung).....	3
Copyright	3
Download	3
Installationsvoraussetzungen.....	3
Features.....	3
Fehlende Features	3
Verwendete Software	3
Bekannte Probleme	4
Kurzanleitung.....	4
Erstellung von Updates	4
Loganalyse	5
Beschreibung der Anwendungsdialoge	5
FileSystem Exclusion.....	6
Registry Exclusion	7
Path Substitution	8
Reg Substitution	9

AppV2Msi – Rechtliches (Warnung)

Diese Software ist experimentell. Die Resultate sind teilweise funktionsfähig, jedoch sind sehr viele Ausschlüsse und Besonderheiten zu beachten. Verwendet werden sollten aktuell nur .appv Files, die mit einem 32Bit Windows 7 sequenziert wurden und die nur 32 Bit Anwendungen enthalten (64 Bit ist implementiert jedoch nicht vollständig ausgetestet). Die Resultate funktionieren dann Teilweise auch unter Windows 7 64 Bit und Windows 8. Die Installation und Deinstallation der MSI Files verlief in einigen Fällen unsauber. Daher bitte nur in Testumgebungen verwenden. Die Nutzung erfolgt in Ihrer eigenen Verantwortung. Also nur etwas für Experten! Wir übernehmen keinerlei Haftung für die Nutzung dieser Software. Wir unterstützen gerne bei Fragen.

Copyright

Die Rechte zu Appv2msi liegen ausschlich bei der Nick Informationstechnik GmbH. Die Nutzung dieser Version (1.0.0.0) ist frei. Wir übernehmen keinerlei Gewährleistung für eventuelle Schäden oder Datenverlust durch die Nutzung dieser Software. © 2013 Nick Informationstechnik GmbH / Andreas Nick

Download

AppV2msi wird ausschließlich über die Webseite <http://www.software-virtualisierung.de> bereitgestellt. Eine Verlinkung ist erwünscht, jedoch nicht die Bereitstellung von Kopien. Die Bereitstellung einer Kopie verstößt gegen unsere Rechte und wird rechtlich verfolgt.

Für die Nutzung und den Download ist eine Anmeldung auf der Webseite <http://www.software-virtualisierung.de> notwendig. Wir gehen sorgsam mit Ihren Daten um, und geben diese nicht weiter. Bitte beachten Sie die Bedingungen bei der Anmeldung zum Thema „Newsletter“ und Kontakt“

Installationsvoraussetzungen

- Windows 7 (32/64 Bit) oder Windows 8/8.1 (32/64 Bit)
- .net Framework 4.0 <http://www.microsoft.com/en-us/download/details.aspx?id=17718>
- Powershell 3.0 <http://www.microsoft.com/en-us/download/details.aspx?id=34595>

Features

- Umsetzung des virtuellen App-V File Systems in ein MSI
- Umsetzung der App-V Registrierung (automatische Konvertierung aller Pfade zu neuem Ziel-Pfad)
- Registrierung und Installation von Fonts (sollte funktionieren)
- Erstellung update fähiger MSIs (weitgehend ungetestet)

Fehlende Features

- Unterstützung von COM (Komponenten werden nicht registriert)
- Unterstützung ROOT Schlüssel (Kontextmenü etc.)
- Assemblies (Assemblies werden nicht erkannt und auch nicht automatisch registriert)
- Shortcuts (weder im Startmenü noch auf dem Desktop werden Shortcuts erstellt. Die Programmierung dafür ist recht leicht und wird eventuell bald nachgereicht)

Verwendete Software

Bitte beachten Sie zu der genutzten Software die entsprechenden Lizenzinformationen

- 7zip <http://7-zip.org/>
- reglookup <http://projects.sentinelchicken.org/reglookup/>
- Windows Installer XML (WiX) Toolset <http://wixtoolset.org/about/license/>

Bekannte Probleme

Fehlermeldung: Key /REGISTRY/USER/{AppVCurrentUserSID} /.....File Folder NONE (null) [0] with name "File Folder" and Value "(null)" in not in the msi

Es gibt einige Registrierungswerte, die als Type eine "Null" haben. Also nicht Reg_sz, DWord o.ä. Hier scheint es keine Möglichkeit zu geben, diese mit WiX zu erzeugen. Hat jemand dazu Informationen?

Kurzanleitung

Zunächst ein neues Projekt anlegen. Der Default-path ist "%appdata%\appv2msi\Projects\". Dort legen Sie ein Verzeichnis für Ihr Projekt fest (Ordner anlegen).

Anschließend über "Select File" ein App-V 5 .appv File auspfählen

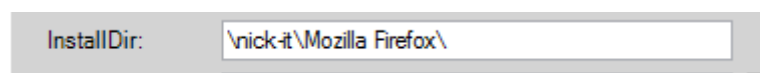
Das Tool versucht aus einigen Konfigurationsdateien (.xml) des .appv Files Informationen zu extrahieren. Das klappt nicht immer vollständig!

Gerade für Packagename, Display Name, Version (x.x.x.x) und Manufacture können beliebige ASCII-Zeichen eingetragen werden.

Bitte besonders den Installationspfad beachten.

Für die Quelle (.appv) darf die Hauptanwendung nur nach c:\program Files (x86)\<APPFOLDER> oder nach c:\Program Files\<APPOLDER> installieren. Der Installationspfad kann angepasst werden. Das Zielverzeichnis (c:\program Files (x86)\ oder c:\program Files\) wird dann nur über den „64“ Bit Schalter gewählt. Ist dieser gesetzt, kann auch eine 32 Bit Anwendung nach c:\Program Files\<APPOLDER> installiert werden.

Beispiel:



Ohne den 64 Bit Schalter landet diese Anwendung unter „c:\program Files (x86)\nick-it\Mozilla-Firefox\“ und das ist korrekt! Also hier diesen Schalter nicht setzen.

Alle Projekteinstellungen werden mit dem Projekt gespeichert und landen in der Datei:

%appdata%\appv2msi\Projects\<MYPROJECT>\<MSINAME>_Settings.xml

Hier landen auch alle Änderungen für Substitutionen und Ausschlüsse, die für ein Projekt vorgenommen werden.

Erstellung von Updates

Aktuell eine weitgehend ungetestete Funktion. Ein sauberes Update ist abhängig von sauberen Guids. Diese dürfen nicht zufallsgeneriert werden. Also den Schalter "Gen random Guid" aus lassen! Die Guids werden dann auf Basis der Pfadnamen generiert. Das Verzeichnis darf sich nicht ändern! Es muss ein neuer Upgradecode eingetragen werden. Der generierte Productcode und Upgradecode

muss identisch zu der Vorversion sein. Weiterhin muss auch eine neue Versionsnummer eingetragen werden. Bei der Versionsnummer muss das Msi Format beachtet werden. Also 1.0.0.0 wird zu 1.0.1.0. Es werden von dem MSI nur die ersten drei Werte betrachtet. Also mindestens an der dritten Stelle ändern! Die GUIDS werden über den Projektpfad generiert. Der muss aktuell exakt identisch zu der Vorversion sein. Also das gleiche Projektverzeichnis unter dem gleichen Pfad. Das alte Verzeichnis sollten Sie sichern! Bitte in jedem Fall den generierten Productcode überprüfen! Wenn der identisch ist, kann die neue Version generiert werden

ANMERKUNG: Wir planen an dieser Stelle einige Erweiterungen

Loganalyse

Im Projektverzeichnis werden unter "Logs" die folgenden Dateien erstellt:

- AppV2XML.log (Log des Generators. Entspricht der Ausgabe im Dialogfenster)
- <APPNAME>.candle.log (Log des WiX-Candle Generators. Wenn kein MSI erzeugt wurde, bitte zunächst hier nachschauen.)
- <APPNAME>.msi.log (Log des WiX-Light Generator. Fehlermeldungen bei der Erstellung des MSI)
- <APPNAME>.xml (WiX Vorlagendatei, die mit dem Tool appv2msi generiert wurde)

Die Logs enthalten teilweise viele Warnungen und auch Errors, einiges davon wird mit neuen Versionen beseitigt. Einiges kann auch problemlos ignoriert werden.

Ausgabe im AppV2XML.log

IMPORTANT INFO [{}] Value not found in substitution table. Please make a entry : [{}]\...

Hier fehlt eine Substitution in der Registrierung. Diese muss im Projekt ergänzt werden. Beispiel [{}] => [DesktopFolder]

Anmerkung: Wir bitten um Infos, welche Pfade hier möglicherweise als Standard enthalten sein sollten!

Wichtige Links in diesem Zusammenhang:

WiX Burn Built-in Variables:

http://wixtoolset.org/documentation/manual/v3/bundle/bundle_built_in_variables.html

Einige Infos von Microsoft:

[http://msdn.microsoft.com/en-us/library/windows/desktop/dd378457\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/dd378457(v=vs.85).aspx)

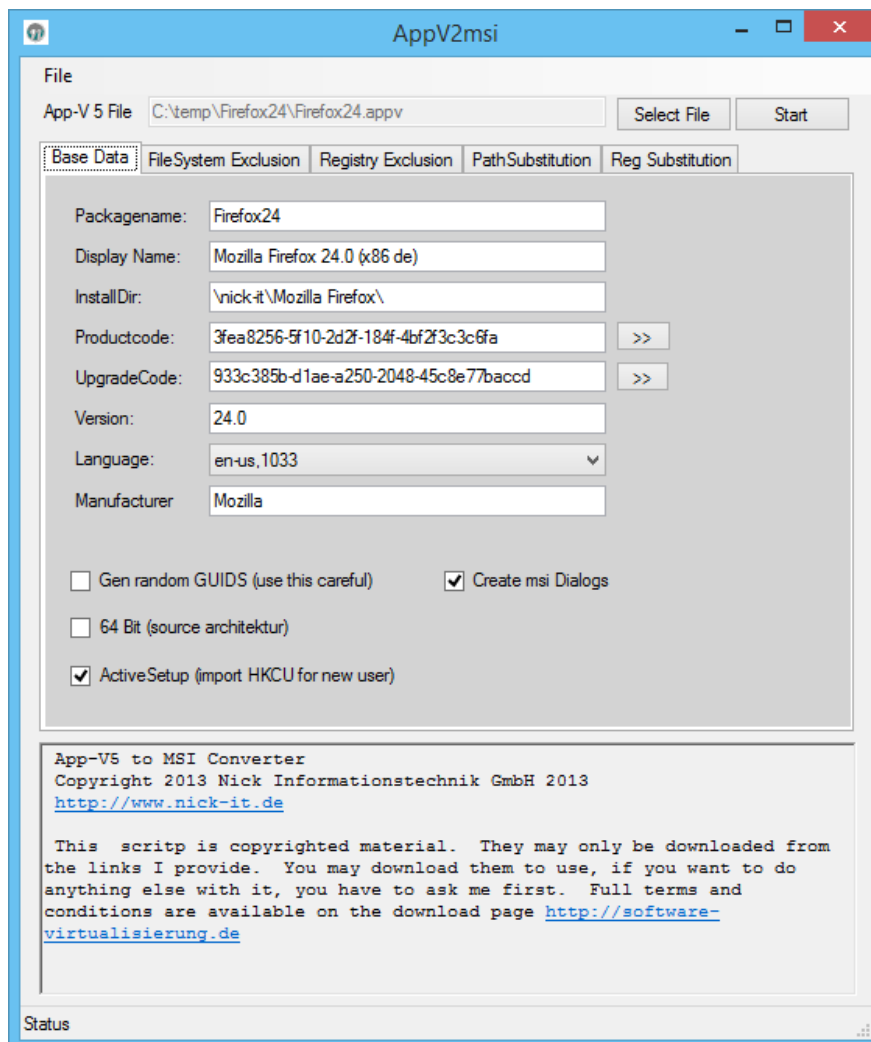
[http://msdn.microsoft.com/en-us/library/windows/desktop/bb762494\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/bb762494(v=vs.85).aspx)

Beschreibung der Anwendungsdialoge

Beschreibung der wichtigsten Felder. Teilweise können die Felder nicht korrekt aus den .appv XML Files ausgelesen werden. Dann müssen diese Daten manuell eingetragen werden

- Packagename wird <Packagename>.msi
- Display Name – Anzeige in der Ansicht der installierten Programme
- Install Dir – Installationspfad unter c:\program Files oder Program Files (x86). Der Pfad kann geändert werden

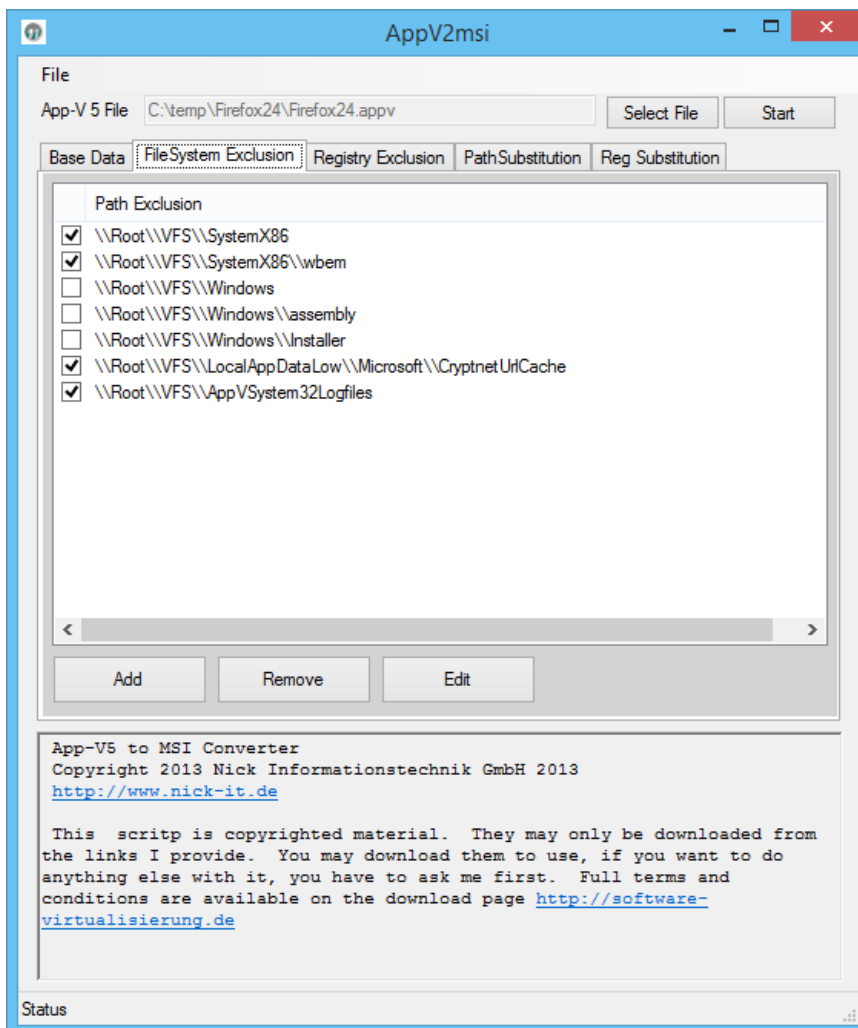
- Version – Bitte hier die MSI notation x.x.x.x beispielsweise 24.0.0.0 eintragen, wenn nicht so ausgelesen
- Gen random GUIDS - erstellt die Systemschlüssel zufallsbasierend. Das kann bei Umfangreichen Anwendungen notwendig werden.
- Create msi Dialogs – Die erstellten msi Files bekommen Dialoge. Diese können unter c:\program Files (x86)\appv2msi\dialogs gerne gegen eigene ausgetauscht werden
- 64 Bit – dieser Schalter muss gesetzt werden, wenn es sich bei der Anwendung um eine 64 Bit Anwendung handelt. Ansonsten wird auch die 64 Bit Anwendung nach c:\Program Files (x86) installiert
- ActiveSetup – es wird der ActiveSetup Schlüssel für neue Anwender gesetzt. Es erfolgt bei der Erstanmeldung ein Reparaturlauf für das msi mit dem ProductCode. Neue Anwender erhalten automatisch alle notwendigen Dateien beispielsweise nach %appdata%. Weiterhin die Registrierungsschlüssel unter HKCU. Diese werden für die Terminalserverkompatibilität nicht deinstalliert.
- Mit „Start“ erfolgt der Build Prozess.



FileSystem Exclusion

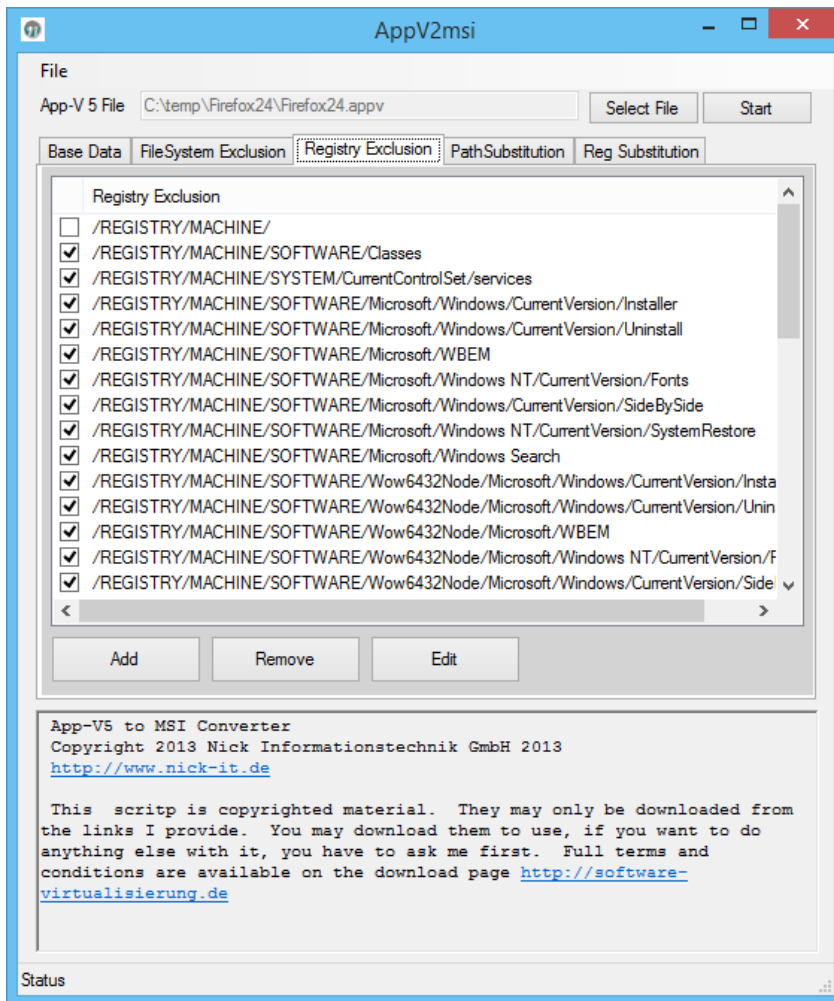
Hier sind Ausschlüsse aus dem Appv Filesystem definiert. Also Ordner, die bei der Konvertierung ignoriert werden. Es werden nur die Ordner berücksichtigt, die markiert sind.

Die die Software mit regulären Ausdrücken arbeitet, ist für ein Backslash jeweils ein Doppeltes zu setzen „\\“. Nach der Durchsicht der Logs hier ggf. erweitern.



Registry Exclusion

Hier sind Registrierungs-Ausschlüsse definiert. Also die Hives, die bei der Konvertierung ignoriert werden. Es werden nur die Zweige berücksichtigt, die markiert sind.

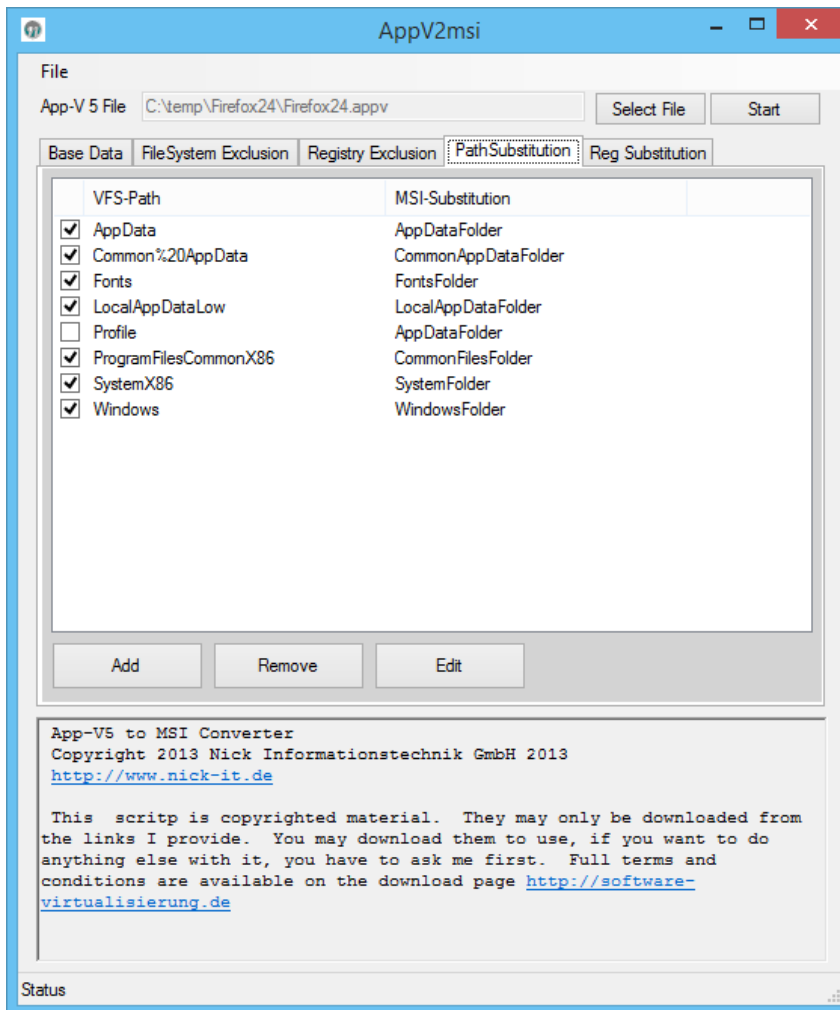


Path Substitution

Es werden die AppV Variablen in msi Variablen gewandelt. Hier muss ggf. nach Durchsicht der Logs ergänzt werden. Eine Besonderheit ist, dass nur die root\vfs Pfade in das msi wandern, die markiert wurden. Alle anderen VFS Verzeichnisse werden ignoriert! Einige Projekte werden hier erhebliche Erweiterungen benötigen. Eine Übersicht findet sich in der WiX Dokumentation:

http://wixtoolset.org/documentation/manual/v3/bundle/bundle_built_in_variables.html

Die AppV Pade werden so angegeben, wie diese im Filesystem zu sehen sind. Also Leerzeichen werden durch ein %20 ersetzt etc.



Reg Substitution

Wandelt die AppV Variablen in der Registrierung zu msi Variablen. Bitte einfach mal in die erweiterten Einstellungen des Sequencers schauen. Davon gibt es eine ganze Menge. Wir haben hier nur die wichtigsten eingetragen. Fehlende Variablen werden in dem Log gemeldet. Gewandelt werden nur die maskierten Variablen.

