

Windows Package Tool (WinPKG)

Version 1.03 – Stand: 17.12.2016
Autor: Patrick Kuhnke
www.Sereby.org

Windows Package (WinPKG) Dokumentation

Versionsübersicht

Version	Bearbeiter	Datum	Bemerkung/Änderungen
1.00	Patrick Kuhnke	05.12.2015	Erstanlage
1.01	Patrick Kuhnke	22.01.2016	Neuer Parameter %arch%
1.02	Patrick Kuhnke	20.08.2016	Dokumentation zu neuen Startparametern
1.03	Patrick Kuhnke	17.12.2016	Neuer Parameter Titel in der WinPKG.xml

Inhaltsverzeichnis

1.	Zielsetzung und System Voraussetzungen	3
2.	Paket-Struktur	3
3.	Aufbau der package.xml.....	3
3.1.	<package> Header	4
3.2.	Variablen	4
3.2.1.	Globale Variablen	4
3.2.2.	<variable> Option	5
3.3.	<cmd> Befehl.....	5
3.4.	<check> Befehl	6
3.4.1.	Definition	6
3.4.2.	Mögliche Optionen.....	7
4.	WinPKG.exe	8
4.1.	Konfiguration von WinPKG	8
4.2.	Startparameter von WinPKG	8
4.3.	Changelog	9

1. Zielsetzung und System Voraussetzungen

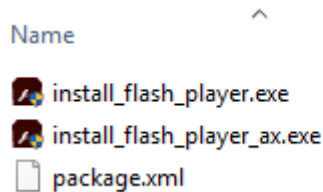
Das Windows Package Tool soll eine einfache Möglichkeit bereitstellen Software auf einem Windows Gerät in der aktuellsten Version zu Installieren. Dabei soll nur noch die Software installiert werden, welche noch nicht auf dem aktuellen Stand ist. Des Weiteren sollen die Pakete einfach zu aktualisieren und erweiterbar sein.

Als Basis für diese Dokumentation wird ein frisch installiertes Windows 7 verwendet. Die genutzte Edition (Home, Ultimate, Enterprise) ist dabei nicht von Bedeutung. Die Software ist getestet unter Windows XP bis hin zu Windows 10 und auf jedem System ohne weitere Runtimes wie z.B. „.NET Framework“ oder „Java“ lauffähig.

2. Paket-Struktur

Beginnend ab dem Verzeichnis in dem die WinPKG.exe liegt wird in jedem Verzeichnis, inklusive aller Unterverzeichnisse, nach der Datei „package.xml“ gesucht. Diese XML Datei definiert die Paket-Informationen wie z.B. Installationsabläufe und prüfungen ob ein Programm bereits installiert ist. In dem gleichen Verzeichnis befinden sich idealerweise auch die Installations-Dateien für das Paket. Diese können aber auch in Unterverzeichnissen hinterlegt werden.

Beispiel für den Adobe Flash Player:



3. Aufbau der package.xml

Eine package.xml Datei für den Adobe Flash Player ist wie folgt aufgebaut und kann als Beispiel genommen werden.

```
1 <package id="AdobeFlash" name="%NAME% v%VERSION%" group="Standard Programme">
2   <variable name="VERSION" value="19.0.0.245" />
3   <variable name="VERSION_FILE" value="19_0_0_245" />
4   <variable name="VERSION_FILE_AX" value="19_0_0_245" />
5   <variable name="NAME" value="Adobe Flash Player" />
6
7   <install>
8     <cmd name="%NAME% - ActiveX" path="install_flash_player_ax.exe" param="-install -au 0">
9       <check type="os" condition="<" value="6.2.9200.0" />
10      <check type="file" condition="notexist" path=
11        "%Windir%\system32\Macromed\Flash\Flash64_%VERSION_FILE_AX%.ocx" arch="x64" />
12      <check type="file" condition="notexist" path=
13        "%Windir%\system32\Macromed\Flash\Flash32_%VERSION_FILE_AX%.ocx" arch="x86" />
14    </cmd>
15    <cmd name="%NAME% - Plugin" path="install_flash_player.exe" param="-install -au 0">
16      <check type="file" condition="notexist" path=
17        "%Windir%\system32\Macromed\Flash\NPSWF64_%VERSION_FILE%.dll" arch="x64" />
18      <check type="file" condition="notexist" path=
19        "%Windir%\system32\Macromed\Flash\NPSWF32_%VERSION_FILE%.dll" arch="x86" />
20    </cmd>
21  </install>
22 </package>
```

3.1. <package> Header

Jedes Package braucht eine einmalige ID, einen verständlichen Namen und einen Gruppennamen unter der das Paket gelistet werden soll. Diese Informationen stehen in der ersten Zeile der package.xml.

Folgende Optionen sind möglich bzw. Erforderlich

NAME	BESCHREIBUNG	DEFAULT
ID	Einmalige ID für das Paket damit intern damit gearbeitet werden kann	Pflichtangabe
NAME	Dies ist der Titel des Paketes der in der Oberfläche angezeigt wird	Pflichtangabe
GROUP	Name der Paketgruppe in der das Paket angezeigt werden soll. Groß/Kleinschreibung sind wichtig.	Pflichtangabe
PRIORITY	Erlaubt sind Zahlenwerte. Diese Option ermöglicht es Pakete vor anderen zu installieren, wenn dies benötigt wird. Je kleiner der Wert desto früher wird das Paket abgearbeitet.	Optional. Standard: 50
DESC	Beschreibung für das Paket.	Optional
PREREQ	Benötigt das Paket zwangsweise ein anderes Paket das zunächst Installiert werden muss, dann kann dessen ID hier angegeben werden.	Optional
CHECKED	Ermöglicht es das Paket nicht automatisch Aktiviert zu initialisieren.	Optional (0/ 1)

Beispiel:

```
<package id="AdobeFlash" name="%NAME%" group="Standard Programme" checked="0" >
```

3.2. Variablen

3.2.1. Globale Variablen

Es gibt bestimmte Variablen die schon Vordefiniert sind und global an jeder Stelle in der package.xml genutzt werden können. Es können auch auf System-Variablen zugegriffen werden wie z.B. %windir%

Folgende Variablen sind möglich

NAME	BESCHREIBUNG	DEFAULT
APP	Gibt den Pfad zu der WinPKG.exe wieder	-
PACKAGE	Gibt den Pfad zu dem Ordner der genutzten package.xml wieder	-
ARCH	Gibt die Architektur des aktuellen Systems wieder. (x86, x64)	-

Windows Package (WinPKG) Dokumentation

3.2.2. `<variable>` Option

Es können häufig genutzte Strings als Variablen definiert werden. In diesem Beispiel wird die Variable „Name“ mit dem Inhalt „Adobe Flash Player“ angelegt. Alle Vorkommnisse in der package.xml die %NAME% enthalten werden dann ersetzt. Diese Option wird nach dem Paket Header definiert.

Folgende Optionen sind möglich bzw. Erforderlich

OPTION	BESCHREIBUNG	DEFAULT
NAME	Name der Variable	Pflichtangabe
VALUE	Wert der Variable	Pflichtangabe

Beispiel:

```
<package id="AdobeFlash" name="%NAME%" group="Standard Programme">
  <variable name="NAME" value="Adobe Flash Player" />
</package>
```

3.3. `<cmd>` Befehl

Innerhalb der Sektion “<install></install>” wird jeder Befehl (`<cmd .. />`) der Reihe nach abgearbeitet

Folgende Optionen sind möglich bzw. Erforderlich

NAME	BESCHREIBUNG	DEFAULT
DEBUG	Option um beim Initialisieren des Paketes die aufgelösten Werte von Variablen auszugeben bei dem <code><cmd></code> Befehl.	Optional (0 /1)
ARCH	x86: der Befehl wird nur auf 32 bit Systemen ausgeführt. x64: der Befehl wird nur auf 64 bit Systemen ausgeführt. *: der Befehl wird immer ausgeführt (Standard).	Optional (x86, x64, *)
PATH	Dateiname des Tools das ausgeführt werden soll.	Pflichtangabe
PARAM	Parameter die zur Ausführung des Setups notwendig sind.	Optional
NAME	Name des Programms. Standardangabe ist der Paketname der im <code><package></code> Header angegeben wurde.	Optional (Paketname)

Beispiel:

```
<package id="AdobeFlash" name="%NAME%" group="Standard Programme">
  <variable name="NAME" value="Adobe Flash Player" />
  <install>
    <cmd name="%NAME% - ActiveX" path="install_flash_player_ax.exe" param="-install -au 0"/>
  </install>
</package>
```

3.4. <check> Befehl

3.4.1. Definition

Es können mehrere Checks nötig sein um zu unterbinden, dass ein Programm installiert wird da das Programm entweder bereits installiert ist oder eine Voraussetzung nicht erfüllt ist. Sollte ein <cmd> durch einen oder mehrere Checks nicht zugelassen werden, dann wird es gar nicht erst zur Installation angeboten.

Die Checks müssen nicht an einen <cmd> gebunden sein, sondern können noch vor dem <install> zweig definiert werden und so schon im Vorwege für das gesamte Paket ein Ausschluss zu definieren.

Beispiel: Ein Paket enthält mehrere Komponenten die installiert werden müssen, dann erhält jeder <cmd> einzeln seine Checks die dafür sorgen, dass diese eventuell nicht installiert werden.

Wenn jedoch vor <install> ein Check auf das Betriebssystem oder eine Datei definiert wurde und dieser nicht erfolgreich ist, dann wird das Paket gar nicht erst weiterbearbeitet und somit auch nichts installiert.

```
<package id="jre8" name="%NAME%" group="Standard Programme" priority="10">
  <variable name="NAME" value="Java Runtime Environment 8 Update 66" />
  <variable name="BUILD" value="8.0.660.18" />
  <variable name="DIR" value="jre1.8.0_66" />

  <check type="os" condition=">=" value="5.1.2600.5512" />
  <check type="file" condition="<" arch="x86" value="%BUILD%"
path="%programfiles%\Java\%DIR%\bin\java.exe" />
  <check type="file" condition="<" arch="x64" value="%BUILD%"
path="%programfiles(x86)%\Java\%DIR%\bin\java.exe" />

  <install>
    <cmd name="Java: Fix MS SystemUser Bug" path="..\SystemFix.bat" />
    <cmd name="Java: Remove old Installation" path="..\uninstall.bat" param="8" />
    <cmd path="x86\jre.exe" param="/s" />
    <cmd path="x64\jre.exe" param="/s" arch="x64" name="%NAME% x64" />
  </install>
</package>
```

3.4.2. Mögliche Optionen

Es ist möglich mehrere `<check .. />` in ein `<cmd ..> </cmd>` zu setzen und diese zu kombinieren. Entweder ein Check für jede Architektur oder es soll erst geprüft werden ob das Betriebssystem erlaubt ist und danach ob die Datei schon existiert (oder nicht). Erst wenn alle Checks erfolgreich waren wird das cmd zur Installation freigegeben.

Folgende Optionen sind möglich bzw. Erforderlich

NAME	BESCHREIBUNG	DEFAULT
TYPE	Definiert die Art der Prüfung <ul style="list-style-type: none">- OS: Prüft auf die Betriebssystem-Version- File: Prüfung auf existenz von Dateien	Pflichtangabe
CONDITION	Je nach zuvor definiertem Type bestehen andere Möglichkeiten. OS: <ul style="list-style-type: none">- *: Alle Versionen erlaubt (Standard)- Vergleichsoperatoren<ul style="list-style-type: none">o <= : Kleiner oder gleicho >= : Größer oder gleicho = : exakt gleicho < : kleiner also > : größer als File: <ul style="list-style-type: none">- exist: Wenn die Datei existiert, dann ..- notexist: Wenn die Datei nicht existiert, dann ..- md5: Wenn der Prüfwert der bei PATH angegebenen Datei exakt mit der bei VALUE übereinstimmt- Standard: Produkt-Version der bei PATH angegebenen Datei prüfen. Die bei OS definierten Vergleichsoperatoren können auch hier angewendet werden	Pflichtangabe
DEBUG	Option um beim Initialisieren des Paketes die aufgelösten Werte von Variablen auszugeben bei dem <code><check></code> Befehl.	Optional (0 /1)
PATH	Dateiname des Tools das ausgeführt werden soll.	Pflichtangabe
VALUE	Mögliche Werte: Versionsnummern und MD5 Werte	Optional
ARCH	x86: der Check wird nur auf 32 bit Systemen ausgeführt x64: der Check wird nur auf 64 bit Systemen ausgeführt *: der Check wird immer ausgeführt (Standard)	Optional Optional (x86, x64, *)

Beispiele:

```
<check type="os" condition="<" value="6.2.9200.0" />
```

```
<check type="file" condition="notexist"  
path="%Windir%\system32\Macromed\Flash\Fash64_%VERSION_FILE%.ocx" arch="x64" />
```

```
<check type="file" condition="<"  
path="%Windir%\system32\Macromed\Flash\Fash32_%VERSION_FILE%.ocx" value="%VERSION%"  
arch="x86" />
```

4. WinPKG.exe

4.1. Konfiguration von WinPKG

WinPKG sucht beim Start eine Konfigurationsdatei namens „WinPKG.xml“ und verarbeitet einige Einstellungen sofern die Datei gefunden wird.

Zunächst wird im Startverzeichnis in der die Anwendung liegt gesucht. Sollte dort nichts gefunden werden wird in %systemdrive% und danach in %windir% und %windir%\system32 gesucht.

Folgende Optionen sind möglich bzw. Erforderlich

OPTION	BESCHREIBUNG	DEFAULT
MAXIMIZED	Maximiert das Programmfenster	Optional (0/1)
REBOOTWHENFINISH	Startet das System am Ende aller Installationen sofort neu.	Optional (0/1)
TITLE	Setzt den Titel im Hauptfenster. Die Zeichenfolge „© by Sereby.org“ wird automatisch angefügt	Optional
<TIMER />	Der integrierte Timer kann über folgende Optionen gesteuert werden. <ul style="list-style-type: none">- Value: Zeit in Sekunden die der Timer runterzählt. Bei „0“ Beginnt die Installation sofort.- Enabled: (0/1) Timer komplett aktivieren oder deaktivieren	Optional
<SET />	Überschreibt den vordefinierten Wert der Option „checked“ aus der angegebenen Package-ID aus den package.xml Dateien. Beide Parameter sind Pflichtangaben. <ul style="list-style-type: none">- component: Package-ID- checked: Paket aktivieren oder deaktivieren (0/1)	Optional

Beispiel:

```
<settings maximized="1" RebootWhenFinish="1" title="Meine Anwendung" >  
  <timer value="99" enabled="0" />  
  <set component="paintnet" checked="0" />  
</settings>
```

4.2. Startparameter von WinPKG

WinPKG.exe kann mit folgenden Parametern gestartet werden.

OPTION	BESCHREIBUNG
/DEBUG	Mit diesem Parameter werden an gewissen Schlüsselpunkten beim Laden der Anwendung und vor dem Start von Installationen Informationen ausgegeben um mögliche Fehler in Paketen ausfindig zu machen.

Windows Package (WinPKG) Dokumentation

/SILENT

Unterdrückt die Meldung, dass keine weiteren Pakete zur Installation zur Verfügung stehen.

4.3. Changelog

Datum	Version	Änderungen
17.12.2016	1.0.0.13	- Neuer Parameter Title in der WinPKG.xml
20.08.2016	1.0.0.12	- Neuer Startparameter /silent
22.01.2016	1.0.0.11	- Neue globale Variable %arch% mit den Optionen x86 bzw. x64
05.12.2015	1.0.0.10	- Windows 10 Support beim Erkennen des Betriebssystems - Neue Prüfmöglichkeit: MD5 Werte bei TYPE="file"
14.05.2014	1.0.0.9	- Changelog nicht mehr bekannt