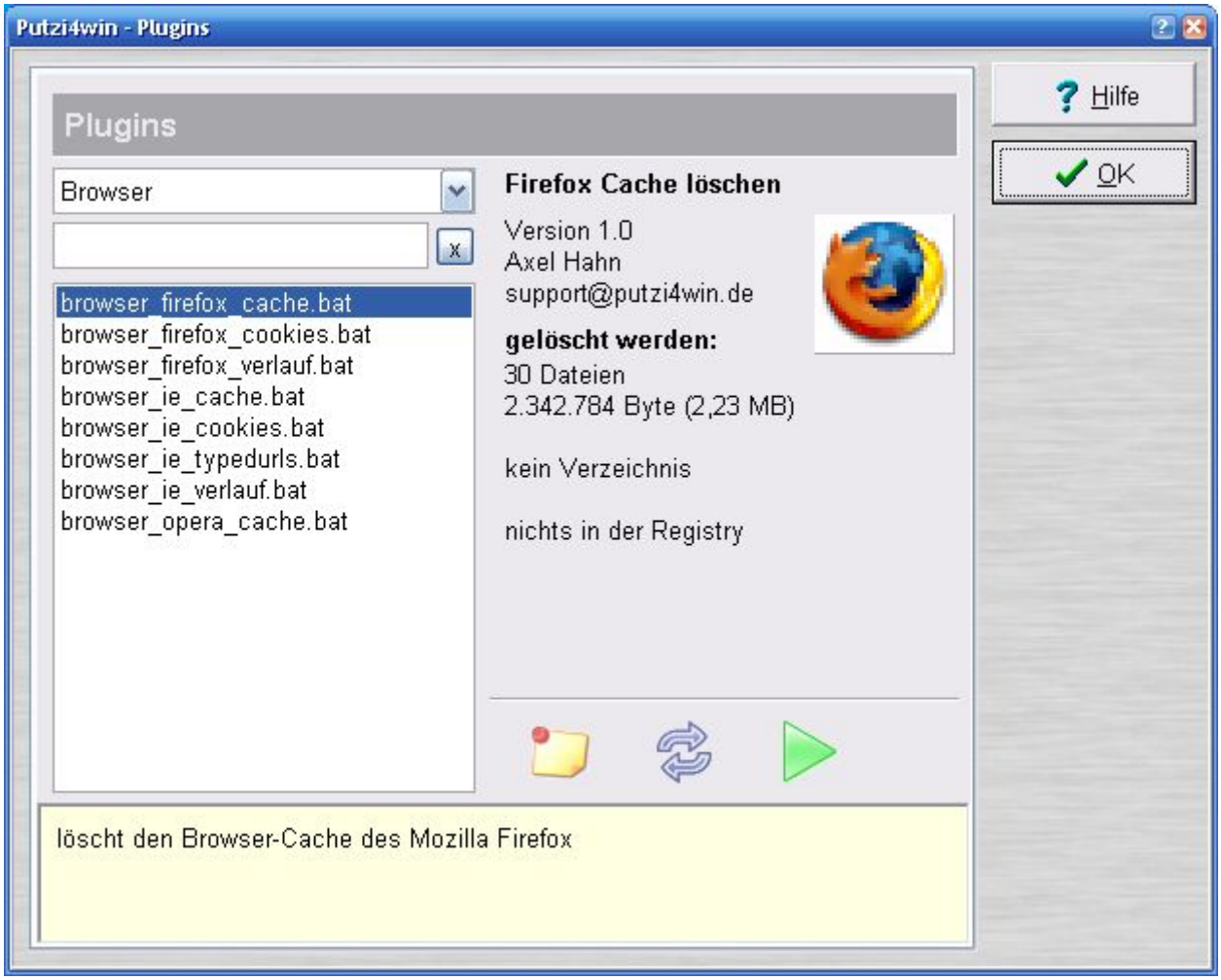
Dokumentation

Putzi4Win

Plugins



http://www.putzi4win.de/
Axel Hahn 2008

Inhaltsverzeichnis:

History dieses Dokuments:

Datum	Version	Was
Bis 02-2007	V0.1	In Arbeit
10-2007	V0.2	Hinzugefügt: shregsv
01-2008	V0.3	Hinzugefügt: MetaWARNING
03-2008	V0.4	Kleinere Korrekturen; Abschnitt “Alle wichtigen Shell-Ordner in Variablen holen” eingefügt
08-2008	V1.0	Hinzugefügt: Icon, Filter für set_dirs.bat

1. Putzi4Win – Plugins

1.1. Wie funktioniert es?

Das Hauptprogramm von Putzi4Win startet ein Plugin wie ein normales Programm und parst anschliessend dessen Ausgabe.

Das Plugin selbst ist ein Skript, das Informationen zusammenstellt und in die Standard-Ausgabe schreibt. Alle ausgegebenen Informationen sind in einer vorgegebenen Syntax an die Standardausgabe zu senden.

Derzeit können Batch-Dateien und VBScripte zum Schreiben von Plugins verwendet werden.

1.2. Vorgaben

- Aufgabe des Plugins ist es hauptsächlich, die Lösch-Informationen lediglich zusammenzutragen und an das Hauptprogramm zu übergeben.
- Das Plugin muss selbst Fehler abfangen – die Fehlermeldung kann an das Hauptprogramm übergeben werden.
- Plugins sind Skripte, die Informationen in einer vorgegebenen Syntax in die Standard-Ausgabe schreiben müssen.
- Die Ausgabe muss eine festgelegte Syntax einhalten. bestehen aus je einer Sektion mit
    - Meta-Angaben
      wie Name und Beschreibung des Plugins und Autor
      und
    - Informationen, welche Einträge gelöscht werden können. Dabei werden unterschieden:
        - Dateien
        - Ordner
        - Registry-Werte (noch nicht unterstützt)
        - Regisry-Schlüssel (noch nicht unterstützt)
    - Es können zur verbesserten Lesbarkeit Kommentare und Leerzeilen eingefügt sein.
- Die Skripte müssen (aus Performance-Gründen) einen Parameter /meta unterstützen. Wird dieser Parameter mitgegeben, dann soll das Plugin die Metadaten und noch nicht die zu löschenden Einträge zurücksenden.
- Das Skript muss selbständig durchlaufen und beenden – es darf keine Benutzer-Interaktion ausgeführt werden.
- Es kann eine BMP-Grafik gleichen Namens bis 64x64 Pixel mitgeliefert werden.

2. Schlüsselwörter in der Ausgabe des Plugins

Die Schlüsselwörter müssen ab Zeilenanfang beginnen. Die Schlüsselwörter sind casesensitiv.

2.1. Meta-Daten

Metadaten sind allgemeine Informationen zum Plugin und dessen Autor.

Schlüsselwort	Beschreibung
metaNAME=*Name des Plugins*	
metaVERSION=*Versionsnr. des Plugins*	
metaAUTHOR=*Name des Autors*	
metaDESCRIPTION=*Beschreibung*	
metaCATEGORY=*Kategoriename*	
metaURLINFO=*Adresse*	noch nicht unterstützt
metaURLDOWNLOAD=*Adresse*	noch nicht unterstützt
MetaEMAIL=*Adresse*	
MetaWARNING=*Text*	Wenn das Löschen der vom Plugin gelieferten Daten nicht 100%-ig sicher ist, kann man einen Text zurückliefern. Wenn man den Eintrag leer lässt, wird das Löschen der Daten als sicher angenommen.

Tabelle 1: Schlüsselwörter der Meta-Daten

2.2. Lösch-Informationen

Schlüsselwort	Beschreibung
FILE:*Dateiname*	Dateiname einer überflüssigen Datei. Sie darf nicht in Hochkomma eingeschlossen sein. Gross- und Kleinschreibung sind – wie bei Windows üblich – egal.
DIR:*Verzeichnisname*	Name eines überflüssigen Ordners. Er darf nicht in Hochkomma eingeschlossen sein. Gross- und Kleinschreibung sind – wie bei Windows üblich – egal. Ein Ordner kann nur gelöscht werden, wenn dieser keine Einträge (Dateien und Unterordner) mehr besitzt.
REGVALUE:*Registry-Wert*	Name des zu löschenden Registry-Wertes.
REGKEY:*Registry-Key*	Der angegebene Schlüssel wird samt existierender Werte und Unterschlüssel gelöscht

Tabelle 2: Schlüsselwörter der Löschinformationen

2.3. Sonstiges

Schlüsselwort	Beschreibung
ERROR:*Text*	Das Plugin muss Fehler selbst abfangen können. Z.B. Wenn eine erforderliche Anwendung nicht installiert oder eine andere Windows-Version erforderlich ist, muss mit ERROR eine Mitteilung an das Hauptprogramm erfolgen. Wenn ein Fehler übergeben wurde, blockiert das Hauptprogramm des Start der Aufräumaktion dieses Plugins (Schalter ist deaktiviert)

Tabelle 3: sonstige Schlüsselwörter

3. Parameter

Mit Hilfe von Parametern sollen verschiedene Aktionen gesteuert werden.

3.1. Ohne Parameter

Es erfolgt die gesamte Ausgabe der Metainformationen als auch der Lösch-Informationen.

3.2. Parameter /meta

Das Zusammenstellen der Löschinformationen kann einige Sekunden dauern. Aus Performancegründen sollen mit dem Parameter /meta lediglich die Kopfinformationen, jedoch nicht die Lösch-Informationen zurückgeliefert werden.

4. Grafik

Für jedes Plugin kann eine Grafik eingeblendet werden. Hier gelten folgende Vorgaben:

- Grafik-Format: BMP
- Grösse: max. 64x64 Pixel
- Dateiname: [Name des Plugins].bmp
- Ablageort: im selben Verzeichnis, wie das Plugin selbst

Beispiel:
Plugin: [Programmverzeichnis]\plugins\browser_firefox_cache.bat
Icon: [Programmverzeichnis]\plugins\browser_firefox_cache.bat**.bmp**

5. Beispiel-Ausgabe

So kann die Ausgabe eines Plugins aussehen:

```
# ------------------------------------------------------------------------
# METADATA
# ------------------------------------------------------------------------
metaNAME=Temp löschen
metaVERSION=0.1
metaAUTHOR=Klaus Mustermann
metaDESCRIPTION=löscht alles aus dem Verzeichnis %TEMP%
metaCATEGORY=Demo
metaURLINFO=
metaURLDOWNLOAD=
metaEMAIL=irgendwer@example.com
metaWARNING=

# ------------------------------------------------------------------------
# delete files in TEMP
# ------------------------------------------------------------------------
FILE:C:\Temp\datei_1.tmp
FILE:C:\Temp\verzeichnis_a\datei_2.tmp
FILE:C:\Temp\verzeichnis_b\datei_3.tmp

# ------------------------------------------------------------------------
# delete subdirs in TEMP
# ------------------------------------------------------------------------
DIR:C:\Temp\verzeichnis_a
DIR:C:\Temp\verzeichnis_b

# ------------------------------------------------------------------------
# END OF PLUGIN
# ------------------------------------------------------------------------
```

Die Trennlinien und mit # beginnenden Kommentare sind zur verbesserten Lesbarkeit eingefügt. Im Programm Putzi4Win erfolgt die Ausgabe mit Syntaxhighlight.
Die Art und Weise, WIE diese Ausgabe erzeugt wird, bleibt Ihnen überlassen: es kann ein statischer Text sein oder (wie in den meisten Fällen) eine dynamische Ausgabe. In einer Batch-Datei erzeugt man diese Ausgabe mit dem echo-Kommando – unter VBScript mit Wscript.echo.

6. Beispiel-Plugins

6.1. Batch-Datei

Aus Performance-Gründen sollten Batch-Dateien einer Skriptsprache vorgezogen werden. Die Skripte sind zumeist kürzer, da keine Objekte und Variablen deklariert werden. Auch das Ausführen von Aktionen ist mit wesentlich weniger (und zuweilen unverständlicherem) Code verbunden. Ein grosser Nachteil ist auch, dass man Batch-Dateien nicht so gut debuggen kann.

Hinweis zur Ausführung:

Damit man sich nicht um die Umwandlung von Sonderzeichen und Umlauten kümmern muss, wird durch das Programm Putzi4Win vor dem Start des Batch-Plugins die Codepage 1252 eingestellt.

Hier nun ein einfaches Beispiel zum Löschen des Temp-Verzeichnisses. TEMP ist eine benutzerabhängige Umgebungsvariable. In einer Batch-Datei kann man auf deren Inhalt mit %TEMP% zugreifen – die Variable wird also vom Prozentzeichen links und rechts eingeschlossen. Die Ausgabe des Inhalts eines Verzeichnisses erfolgt mit dem Kommando DIR. Dieses Kommando versteht eine Reihe von Parametern – um die Liste zu sortieren, rekursiv, in Kurz- oder Langform darzustellen uvm. Rufen Sie die Windows-Hilfe auf oder geben Sie *help dir* in der Kommandozeile ein, um weitere Details zum Kommando zu erfahren.

Zur Arbeitsweise des Plugins:

1. Das Plugin gibt mit echo zunächst die Metadaten aus.
2. Mit
   if "%1"=="/meta" goto end
   wird der übergebene erste Parameter geprüft – ist dessen Inhalt “/meta”, wird zum Label “end” (unten im Skript) gesprungen.
3. Die Zeile
   if "%TEMP%"=="" echo ERROR:... && goto end
   testet den Inhalt der Umgebungsvariable TEMP – für den Fall, dass sie leer ist, wird eine Fehlermeldung (Keywort ERROR:) ausgegeben und zum Ende des Skripts esprungen.
4. Die beiden FOR-Schleifen loopen jeweils über die Ausgabe des DIR-Kommandos (dieses listet einmal Dateien und dann Ordnernamen auf) – es wird das Keywort für eine Datei bzw. einen Ordner davorgesetzt:
   for /F "TOKENS=" %%a in ('dir /a:-D /s /b %TEMP%') do echo FILE:%%a*
   (...)
   for /F "TOKENS=" %%a in ('dir /a:D /b %TEMP%') do echo DIR:%TEMP%\%%a*

Und hier der vollständige Code:

```
@echo off
rem ###########################################################################
::
::     Putzi4win-Plugin - TEMP-Verzeichnisse löschen
::
rem ---------------------------------------------------------------------------
::     2006-09-18   support@putzi4win.de
rem ###########################################################################

rem ---------------------------------------------------------------------------
::     CONFIG
rem ---------------------------------------------------------------------------

       set commentLine=#-------------------------------------------------------------------------


rem ---------------------------------------------------------------------------
::     SHOW METADATA
rem ---------------------------------------------------------------------------
       echo %commentLine%
       echo # METADATA
       echo %commentLine%
       echo metaNAME=Temp löschen
       echo metaVERSION=0.1
       echo metaAUTHOR=Axel Hahn
       echo metaDESCRIPTION=löscht alles aus dem Verzeichnis %%TEMP%% und %%TMP%%
       echo metaCATEGORY=Demo
       echo metaURLINFO=
       echo metaURLDOWNLOAD=
       echo metaEMAIL=support@putzi4win.de
       echo metaWARNING=
       if "%1"=="/meta" goto end
       echo.


rem ---------------------------------------------------------------------------
::
::     SHOW DATA
::
rem ---------------------------------------------------------------------------

       if "%TEMP%"=="" echo ERROR:Die Variable TEMP ist nicht gesetzt. Bearbeitung wird
abgebrochen. && goto end

       echo %commentLine%
       echo # delete files in TEMP
       echo %commentLine%

       for /F "TOKENS=*" %%a in ('dir /a:-D /s /b %TEMP%') do echo FILE:%%a
       echo.

       echo %commentLine%
       echo # delete subdirs in TEMP
       echo %commentLine%
       for /F "TOKENS=*" %%a in ('dir /a:D /b %TEMP%') do echo DIR:%TEMP%\%%a
       echo.

       echo.
       echo %commentLine%
       echo # END OF PLUGIN
       echo %commentLine%

       goto end


rem ---------------------------------------------------------------------------
::     END
rem ---------------------------------------------------------------------------
       :end

rem ---------------------------------------------------------------------------
::     EOF
rem ---------------------------------------------------------------------------
```

6.2. Vbscript

Hier noch ein ähnliches (vereinfachtes) Beispiel mit VBScript umgesetzt. Die VBScripte müssen die Erweiterung vbs aufweisen.

Hinweis zur Ausführung:

Alle VBScript-Plugins werden mit der cscript.exe und dem Parameter //nologo ausgeführt.

Arbeitsschritte des Skripts:

1. Mit einem echo werden die Parameter ausgegeben.
2. Es werden die Parameter (im Object Wscript.Arguments) durchgeprüft – wenn ein Parameter /meta ist, wird das Skript beendet.
3. Der Zugriff auf die Umgebungsvariable TEMP erfolgt mit
   set Shell=CreateObject("WScript.Shell")
   Set env=Shell.Environment("PROCESS")
   sDataDir = env("temp")&"\"
4. Der Zugriff der Dateiobjekte wird das Objekt Scripting.FileSystemObject benötigt. Im Beispiel hier werden nur Dateien und diese nicht rekursiv ausgegeben.

Und hier der vollständige Code:

```
' ######################################################################
'
'      DEMO-Plugin - Putzi4Win
'
' ######################################################################


      option explicit
      Dim arg

' ---------------------------------------------------------------------
' show metadata
' ---------------------------------------------------------------------
      wscript.echo "#"
      wscript.echo "# METADATA"
      wscript.echo "#"
      wscript.echo "metaNAME=Testplugin mit VBS"
      wscript.echo "metaVERSION=0.1"
      wscript.echo "metaAUTHOR=Axel Hahn"
      wscript.echo "metaDESCRIPTION=löscht backup-Files von Putzi4Win"
      wscript.echo "metaCATEGORY=Demo"
      wscript.echo "metaURLINFO="
      wscript.echo "metaURLDOWNLOAD="
      wscript.echo "metaEMAIL="
      wscript.echo "metaWARNING="

      For Each arg in Wscript.Arguments
            if (arg = "/meta") then
                  wscript.quit
            end if
      Next


' ---------------------------------------------------------------------
' show data
' ---------------------------------------------------------------------
```

```
	Dim fso,shell, env, sDatadir, f,fc,sc,f1

	Set fso = CreateObject("Scripting.FileSystemObject")
	set Shell=CreateObject("WScript.Shell")
	Set env=Shell.Environment("PROCESS")
	sDataDir = env("temp")&"\"

	Set f = fso.GetFolder(sDataDir)
	Set fc = f.Files

	wscript.echo "#"
	wscript.echo "# DATA"
	wscript.echo "#"
	For Each f1 in fc
		wscript.echo "FILE:" & sDataDir & f1.name
	Next

	wscript.echo "#"
	wscript.echo "# EOF"
	wscript.echo "#"

	wscript.quit

' ---------------------------------------------------------------------------
' EOF
' ---------------------------------------------------------------------------
```

7. Tipps und Tricks

7.1. Plugin-Wizzard

Wenn Sie in einem bestimmten Ordner Daten haben, die Sie ruhigen Gewwissens nach n Tagen löschen können, dann ist der Plugin-Wizzard für Sie vielleicht interessant. Sie können diesen verwenden, um z.B. alte Logdateien zu löschen oder um Ihren Download-Ordner von alten “Leichen” zu befreien.

Der Plugin-Wizzard ist eine HTA-Anwendung, die ein Plugin als Batchdatei nach diversen Vorgaben erzeugt.

Bitte sehen Sie sich die separate Dokumentation der plugin-wizard.pdf an, um eine detaillierte Anleitung zu erhalten.

7.2. Vereinfachung zum Auflisten von Dateien

Sie können die Dateien in Ihrem Plugin manuell auflisten ... z.B. alle Dateien im TEMP-Ordner erhalten Sie mit:

```
for /F "TOKENS=*" %%a in ('dir /a:-D /s /b %TEMP%') do echo FILE:%%a
```

Ebenso in einer FOR-Schleife wurden die Verzeichnisse aufgelistet:

```
for /F "TOKENS=*" %%a in ('dir /a:D /b %TEMP%') do echo DIR:%TEMP%\%%a
```

Wem diese Syntax zu schwierig ist, sollte einmal ein Plugin mit dem Plugin-Wizzard erstellen und sich den Quellcode der erzeugten BAT-Datei ansehen.

Diese benutzt eine Batch-Datei im Unterverzeichnis \plugins\tools\, um Dateien und Unterverzeichnisse aufzulisten. Mit Angabe der Parameter in der vorgegebenen Reihenfolge werden Dateien/ Verzeichnisse angezeigt:

```
set dir="D:\Logfiles\IIS "
set filefilter="ex*.log"
set ageInDays="45"
set recursive=1
set attReadonly=0
set attHidden=0
set emptyDirs=0
(...)
cd /d "%0\..\tools"
call "list_files.bat" %dir% %filefilter% %ageInDays% %recursive% %attReadonly% %attHidden%
%EmptyDirs%
```

7.3. Registry-Wert in eine Variable holen

Manche Ordner kann man nicht aus Umgebungsvariablen auslesen oder gar hart codieren, wenn man ein Plugin auch auf anderen Rechnern laufen soll. Um einen Wert der Registry in eine Variable der Batch-Daei zu holen, können Sie wie folgt vorgehen:

1. In einer Variable setzen Sie den Namen des Schlüssels und den Wert
2. In der FOR-Schleife wird die Kommandozeile mit der shregv.exe zum Lesen des Registry-Wertes ausgeführt (diese befindet sich im Verzeichnis .\plugins\tools\). Das nachfolgende findstr filtert die Ausgabe auf eine einzelne Zeile. Mit Hilfe der Parameter tokens und delims steht in der verwendeten Variable %%i der gewünschte Wert, der mit set der Variable CacheDir zugewiesen wird.

Die Beschreibung in Prosa ist hier einmal umgesetzt – gelesen wird das Cache-Verzeichnis des Internet Explorer:

```
set regValue=HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell
Folders\Cache
(...)
set CacheDir=
set PATH=%PATH%;%~dp0\tools\
For /F "tokens=2* delims==" %%i in ('shregv "%regValue%" ^| findstr "="' ) Do set CacheDir=%%i
```

Anm.:

Die erste Zeile und die Anweisung der FOR-Anweisung sind in eine Zeile zu schreiben.

Öffnen Sie einmal den Registry-Key *HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\ Explorer\Shell Folders*\ - hier lassen sich weitere Systemordner auslesen, etwa der Ordner für das Startmenü, den Desktop oder die lokalen Anwendungsdaten.

Oder beachten Sie den nächsten Abschnitt.

7.4. Alle wichtigen Shell-Ordner in Variablen holen

Ein Bat-Skript, das Werte im Registry-Key HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\ Explorer\Shell Folders\ ausliest und in je eine Variable packt, ist die set_dirs.bat im Ordner plugins\tools\.

Binden Sie in Ihr Plugin diese Zeile ein:

```
call "%~dp0\tools\set_dirs.bat"
```

Anschliessend sind etliche Variablen mit dem Namen DIR_[Name_des_Registrywertes] gesetzt.

Anm.:

Wenn der Registry-Wert ein Leerzeichen enthält, so wird dieses durch ein _ (Unterstrich) ersetzt. Der Wert des Registry-Wertes “Local AppData” ist dann in der Variable “DIR_ Local_AppData” zu finden.

Die Liste der gesetzen Variablen wird als Orientierungshilfe als Kommentar in die Plugin-Ausgabe geschrieben:

```
# ================================================================================
#
# START SET DIRS
#
# setting values of directories from registry key
# HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\
#
# DIR_AppData="C:\Dokumente und Einstellungen\Axel\Anwendungsdaten"
# DIR_Cookies="C:\Dokumente und Einstellungen\Axel\Cookies"
# DIR_Desktop="C:\Dokumente und Einstellungen\Axel\Desktop"
# DIR_Favorites="C:\Dokumente und Einstellungen\Axel\Favoriten"
# DIR_NetHood="C:\Dokumente und Einstellungen\Axel\Netzwerkumgebung"
(...)
#
# ... I hope you can use one or some ;-)
#
```

Der Loop über alle Einträge braucht ca. eine halbe Sekunde – und meist braucht man nicht alle, sondern nur einen der dortigen Ordner. Man kann als Parameter einen Filter angeben. Dieser Filter wird die Namen der Registry-Werte angesetzt und ist nicht case-sensitiv.

Beispiel: Ermitteln des Desktop-Ordners

zum Einbinden schreiben Sie in Ihr Plugin:

```
call "%~dp0\tools\set_dirs.bat" desktop
```

Anschliessend haben Sie mit der Variable DIR_Desktop den Ordner des Desktops des aktuellen Benutzers zur Verfügung.