



We capture the future.

Janich & Klass



scanner value pack

PlugIn zu DpuScan

# Script Plugin

Ergänzung zum DpuScan Referenzhandbuch

## Copyrights

© 1997 bis 2010 Janich & Klass Computertechnik GmbH. Alle Rechte vorbehalten.

Gedruckt in Deutschland.

Die in dieser Dokumentation enthaltenen Informationen sind Eigentum der Janich & Klass Computertechnik GmbH. Ohne schriftliche Genehmigung der Janich & Klass Computertechnik GmbH begründen weder der Empfang noch der Besitz dieser Informationen irgendein Recht auf Reproduktion oder Veröffentlichung irgendwelcher Teile davon.

## Warenzeichen

Das DPU Logo ist eingetragenes Warenzeichen der Janich & Klass Computertechnik GmbH.

DpuScan ist Warenzeichen von J&K Imaging, Marietta/USA. Alle anderen Produktnamen und Logos sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Eigentümer.

## Haftungsausschluss

Die Anweisungen und Beschreibungen in diesem Handbuch waren zum Druckzeitpunkt zutreffend.

Wir behalten uns jedoch das Recht vor, sowohl Beschreibung als auch Produkt jederzeit ohne Benachrichtigung zu ändern.

Nach dem derzeitigen Stand der Softwaretechnik ist es nicht möglich Programme zu entwickeln, die unter allen Bedingungen und in jeder Konfiguration fehlerfrei arbeiten. Die Janich & Klass Computertechnik GmbH übernimmt keinerlei Haftung für Defekte, die direkt oder indirekt durch Fehler dieses Handbuches, Weglassen von Informationen oder durch Unstimmigkeiten zwischen Handbuch und dem Produkt entstanden sind.

## Aktualität

Es ist möglich, dass im Internet eine neuere Version dieser Dokumentation zum DpuScan verfügbar ist. Wir empfehlen deshalb, die Version an Hand des auf dieser Seite abgedruckten Datums mit der Version auf dem Internet zu vergleichen. Falls die Version im Internet neueren Datums ist, sollten Sie diese herunterladen und ggf. selbst ausdrucken.

Die aktuelle Version dieses Anhangs zum DpuScan Referenzhandbuch finden Sie im Web unter:

<http://www.dpuscan.com/pdf/PlugIns/ScriptPlugin-deutsch.pdf>

© 2010 Janich & Klass Computertechnik GmbH, Wuppertal, Germany

23. September 2010

## Inhaltsverzeichnis

1	VBScript.....	4
2	Voraussetzung für die Benutzung von VBScript.....	5
3	Anwendungsfälle für VBScript.....	7
3.1	VBScript in der Taskliste (Process).....	7
3.2	VBScript auf einer Schaltfläche / in Makro (Process) .....	7
3.3	VBScript zur Ereignisbehandlung (Event) .....	7
3.4	VBScript bei Selektionswechsel (Selection) .....	7
4	Gestaltung des VBScript-Programmtextes.....	8
4.1	Interne Funktion.....	8
4.2	Exportierte Funktionen.....	8
4.2.1	Aufruf der Funktionen und Aufrufreihenfolge.....	9
4.2.2	Funktions-Parameter .....	9
4.2.3	Funktions-Rückgabewerte .....	10
4.2.4	Verfügbare Events (wie bei Broker) .....	11
5	Konfiguration des Plugins .....	13
5.1	Selektion von Plugin und Subprofil.....	13
5.2	Eingabe des VBScript Textes .....	14
5.3	Besondere Verfahren in VBScript .....	16
6	Laufzeitfehler, Debugging .....	17

Plugins für DpuScan sind separat zu lizensierende Funktionserweiterungen. Diese Dokumentation beschreibt ein solches Zusatzmodul für bereits bestehende Lizenzen von DpuScan. Die Nutzung des Plugins ist nur zusammen mit DpuScan möglich. Diese Dokumentation ist daher auch nur zusammen mit der Dokumentation von DpuScan nutzbar.

## 1 VBScript

Die Funktionalität von DpuScan kann an vielen Stellen erweitert werden, um spezielle Aufgaben zu bewältigen, die sich aus außergewöhnlichen Anforderungen ergeben. Erweiterungen von DpuScan sind auf Ebene der Ereignisregeln die Function, in der Speicherung der Broker, und auf universeller Ebene für die Behandlung von speziellen Ereignissen und für die Bildbearbeitung das Plugin.

Das Plugin kann auch Broker- und Function-Aufgaben übernehmen.

Zu den speziellen Ereignissen gehören die Brokerereignisse, Benutzeraktionen und sonstige, ganz spezielle Ereignisse des Systemes DpuScan.

Zur Erzeugung eines Plugin ist Programmierung erforderlich. Programmierung verlangt nach einem Entwicklungssystem, einem Compiler und nach speziellen Kenntnissen des Windows-Betriebssystems.

Das spezielle Script-Plugin soll dem Administrator von DpuScan die Möglichkeit geben, **auch ohne Entwicklungssystem** einfache und auch komplexe Aufgaben der Datenverarbeitung zu bewältigen. Das Script-Plugin akzeptiert Programmtext in VBScript und führt dieses Programm unmittelbar aus.

Aufrufe von VBScript-Funktionen

VBScript Funktionen können aufgerufen werden

- als Aufruf von einem Taskschritt
- als Aufruf von einem Schaltfläche in der Benutzeroberfläche
- als Aufruf aus einem Makro
- als Ereignis innerhalb DpuScan, zB bei Bildwechsel, Bilddateispeicherung, Finalisieren

Einschränkungen

- das Script-Plugin kann keine Bildverarbeitung durchführen

Anwendungsbeispiele


- Herausfiltern einer Bestellnummer aus einem OCR-Ergebnis, frei
- Filtern von Zeichenketten über Regular Expressions
- Datenbankzugriffe schreibend, lesend
- Ermitteln einer Stapelnummer aus einem netz-zentralen Nummernspender
- komplexe mathematische Berechnungen

## 2 Voraussetzung für die Benutzung von VBScript

Um VBScript nutzen zu können, muss das Plugin

Script Plugin      JKScrp      JKScriptPlg.dll

in das Basisprofil hineinselektiert werden. Die exportierten Funktionen des VBScripts können dann an den genannten Stellen als Plugin-Aufruf oder auch implizit über Events aufgerufen werden.

Das PlugIn ist innerhalb der Basisprofilkonfiguration zu laden. Öffnen Sie dazu die **Basisprofilkonfiguration**, wählen Sie dort die Registerkarte **Prozess** und klicken Sie auf die Schaltfläche  neben den **Plugins**.

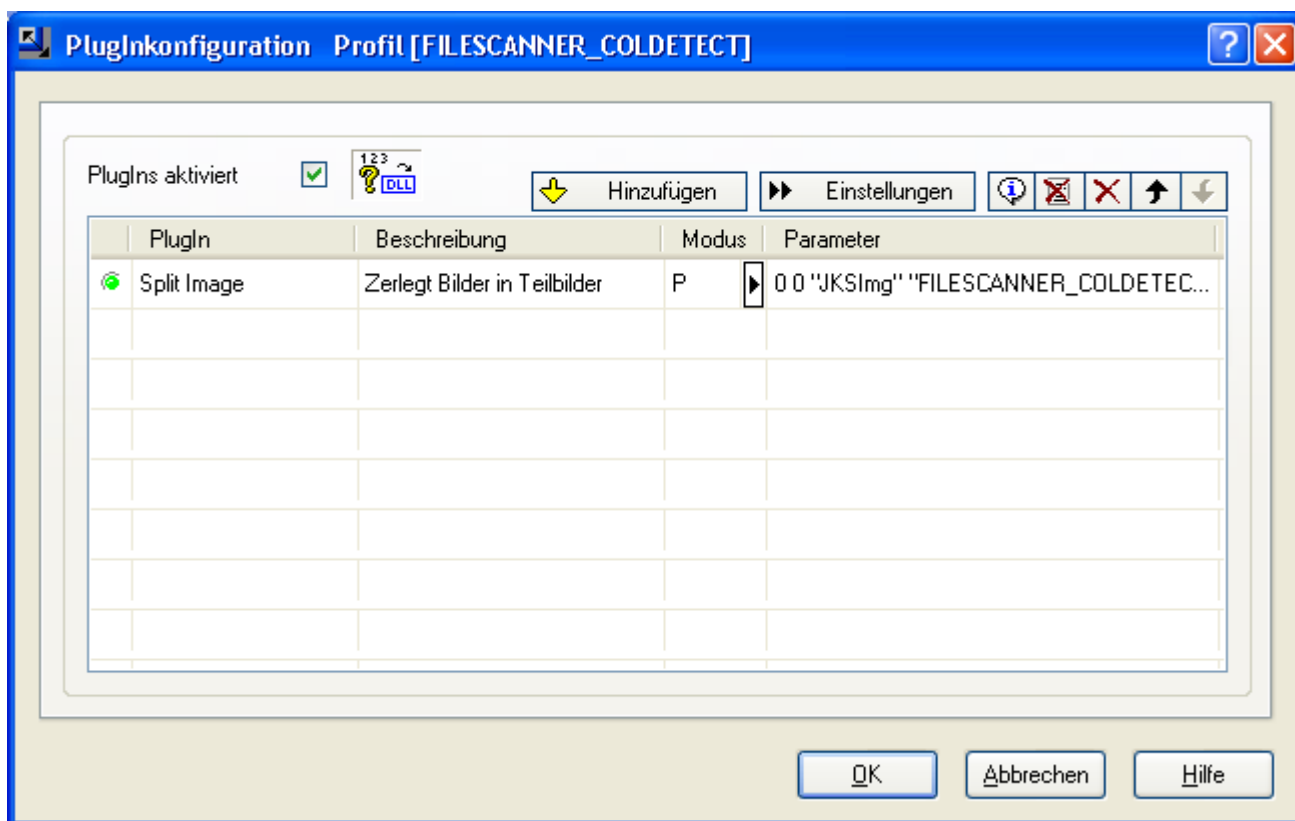


Abbildung 1 – Hinzufügen des Plugins

Die Schaltfläche **Hinzufügen** öffnet die Auswahl verfügbarer Plugins.

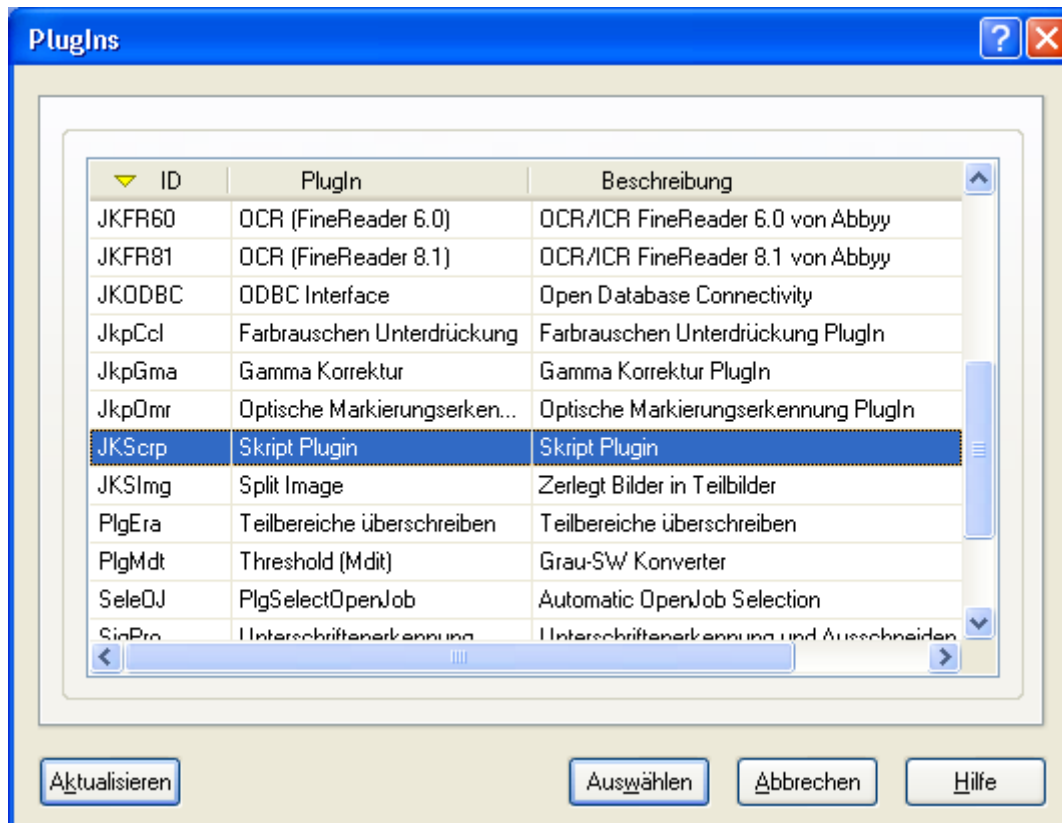


Abbildung 2 – Auswahl des PlugIns

Wählen Sie nun **Script Plugin** aus, indem Sie die entsprechende Zeile selektieren und mit **Auswählen** übernehmen.

In der Pluginkonfiguration verzweigt **Einstellungen** in die Konfiguration des VBScript. Darin können Funktionen zur Behandlung von Ereignissen abgelegt sein. Sind keine solchen Funktionen notwendig kann hier ein beliebiges anderes VBScript oder besser [Voreinstellung] selektiert sein.

## 3 Anwendungsfälle für VBScript

### 3.1 VBScript in der Taskliste (Process)

Ein Aufruf an ein VBScript kann in der Taskliste an beliebige Stellen gesetzt werden. Der Aufruf des VBScriptes erfolgt über den generellen Befehl: **Plugin aufrufen**. DpuScan stellt unterschiedliche Aufrufbefehle bereit für Bereiche der Task, die nur einmal durchlaufen werden, und solche Bereiche, die in der Scanschleife wiederholt aufgerufen werden.

Es können beliebig viele solcher VBScript-Aufrufe eingefügt werden. Ein bestimmtes VBScript trägt einen Namen, wird als Teil der Plugin-Konfiguration gespeichert, kann unter diesem Namen von anderen Plugin-Schritten und auch von anderen DpuScan-Profilen verwendet werden.

### 3.2 VBScript auf einer Schaltfläche / in Makro (Process)

Ein VBScript kann von einer Schaltfläche in der Benutzeroberfläche und aus einem Makro heraus aufgerufen werden. Für beide Anwendungsfälle gelten dieselben Regeln und Konfigurationsmöglichkeiten.

### 3.3 VBScript zur Ereignisbehandlung (Event)

VBScript-Funktionen, die zu DpuScan-Ereignissen – dazu zählen auch die Selections – aufgerufen werden sollen, müssen alle in demselben VBScript definiert sein. Dieses VBScript ist im Basisprofil dem Script-Plugin zugeordnet. Innerhalb des VBScript müssen die Methoden (Function, Sub) zur Behandlung von Ereignissen im Funktionskopf entsprechend gekennzeichnet sein. Weiteres beschreibt Kapitel 4.2

Ereignisse, zu denen VBScript-Funktionen aufgerufen werden, sind z.B.

Task Start

Task Ende

Erzeugen einer Bilddatei

Eine genaue Beschreibung aller Ereignisse liefert Kapitel 4.2 .

Eventhandler nur unter [Basisprofile | Prozess | Plugin | Script Plugin](#)

### 3.4 VBScript bei Selektionswechsel (Selection)

Ein Selektionswechsel ist eine besondere Form des Ereignisses. Es gelten alle Regeln wie für die Ereignisbehandlung. Event-Handler und Selection-Handler müssen im selben VBScript stehen.

Die Selection fängt den Zeitpunkt zu dem das neue Bild bereits selektiert ist:

- alle I-Variablen des neuen Bildes sind verfügbar
- Funktionsrückgaben verändern die I-Variablen des neuen Bildes
- andere Variablen können übergeben oder verändert werden

Ein Ereignis Preselection ist derzeit nicht verfügbar.

Event/Selection-Handler nur unter [Basisprofile | Prozess | Plugin | Script Plugin](#)

## 4 Gestaltung des VBScript-Programmtextes

Im Programmtext von VBScript können fast alle Funktionen und Objekte der VBScript-Runtime benutzt werden. Die Schnittstelle zu DpuScan bilden die **exportierten Funktionen**. Solche Funktionen haben einen speziellen Aufbau, damit DpuScan sie als exportierte Funktionen erkennt, sie auch aufruft und auch Parameter übergibt und Rückgabewerte verarbeitet.

### 4.1 Interne Funktion

VBScript-interne Funktionen, sind solche Funktionen, die nur von innerhalb des VBScript aufgerufen werden. DpuScan ruft solche Funktionen nie direkt auf.

Interne Funktion können wie üblich in VBScript formuliert werden

```
Function IncrCounter (ICounter)
    IncrCounter= ICounter+1
End Function
```

oder auch

```
Private Function IncrCounter (ICounter)
    IncrCounter= ICounter+1
End Function

Sub MyMsgBox (dispText,dispHeader)
    MsgBox dispText,,dispHeader
End Sub
```

### 4.2 Exportierte Funktionen

Exportierte Funktionen sind solche Funktionen im VBScript, die DpuScan direkt aufruft.

Exportierte Funktionen tragen einen speziellen DpuScan-spezifischen Kopf. Exportierte Funktionen sollten grundsätzlich mit dem Funktionsgenerator des Script-Plugins erzeugt werden. Der Kopf der Funktion definiert Übergaberegeln für Parameter und Rückgabewerte.

Exportierte Funktionen tragen im Funktionskopf die Information, zu welchem Zeitpunkt sie aufgerufen werden. Einer der drei Fälle ist möglich

\$\$ MODE Process	Aufruf aus Task, Schaltfläche oder Makro
\$\$ MODE Event=14	Aufruf zu bestimmten DpuScan-Ereignissen
\$\$ MODE Selection	Aufruf nach Bildselektionswechsel

Die Angabe des MODE trägt der Funktionsgenerator ein. Er setzt auch den passenden Zahlenwert für Event ein.

Funktionen der Aufrufmodi Event und Selection müssen im selben VBScript-Programmtext definiert sein und sind nur unter **Basisprofile | Prozess | Plugin | Script Plugin** gespeichert.



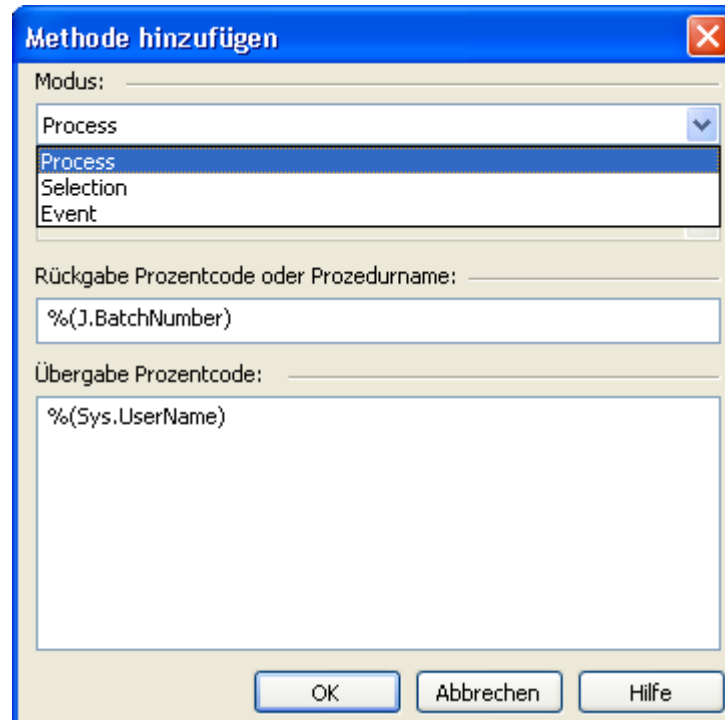


Abbildung 3 – Funktionsgenerator

In einem DpuScan-Profil kann es also

beliebig viele Process-VBScripTs geben

nur ein einziges Event-VBScript geben

#### 4.2.1 Aufruf der Funktionen und Aufrufreihenfolge

DpuScan ruft **alle** exportierten Process Funktionen mit dem Aufruf des Plugins auf.

DpuScan ruft exportierte Funktionen in der Reihenfolge ihres Erscheinens im VBScript-Programmtext auf.

Ereignisfunktionen werden zum festgelegten Ereignis aufgerufen.

#### 4.2.2 Funktions-Parameter

Alle Parameter, die zu exportierten Funktionen gelangen, sind vom Typ String.

VBScript führt einige Wandlungen implizit aus, andere Wandlungen müssen explizit ausgeführt werden:

```
Function Calculate(a)
'$$ DPUSCAN_DECLARATION_BEGIN
'$$ MODE Process
'$$ VAR a = %(I.A)
'$$ VAR Calculate = %(I.Res0)
'$$ DPUSCAN_DECLARATION_END
```

a = a + 1 ' wandelt a in Numeric und erhöht den Wert

Calculate= a ' Rückgabe hier als Numeric, DpuScan wandelt nach String

```
End Function
```

Die Parameter aller Funktionen, die quasi gleichzeitig aufgerufen werden, legt DpuScan vor dem Aufruf der ersten Funktion fest und ändert sie nicht mehr von dem Aufruf der einen exportierten Funktion zum Aufruf der nächsten exportierten Funktion. Eine exportierte Funktion, die weiter oben im Programmtext steht, kann also *nicht* mit ihrem Rückgabewert den Parameter der nächsten exportierten Funktion bestimmen.

#### 4.2.3 Funktions-Rückgabewerte

Eine Funktion gibt genau einen Wert zurück. Der Rückgabewert wird in DpuScan einer DpuScan-Variablen zugewiesen. Sollen mehrere Werte zurückgegeben werden, können mehrere Funktionen geschrieben werden. Die erste Funktion führt die Hauptarbeit aus und speichert ihre Ergebnisse in VBScript-globale Variablen. Weitere Funktionen geben die Werte dieser globalen Variablen zurück.

```
Dim Res1, Res2, Res3

Function QueryDB(a)
'$$ DPUSCAN_DECLARATION_BEGIN
'$$ MODE Process
'$$ VAR a = %(I.A)
'$$ VAR QueryDB= %(I.Res0)
'$$ DPUSCAN_DECLARATION_END
Dim lRes0, lRes1, lRes2

    InvokeSqlQuery(a,lRes0,lRes1,lRes2)
    Res1= lRes1 ' global zwischenspeichern
    Res2= lRes2 ' global zwischenspeichern
    QueryDB= lRes0
End Function

Function QDb_Res1
'$$ DPUSCAN_DECLARATION_BEGIN
'$$ MODE Process
'$$ VAR QDb_Res1= %(I.Res1)
'$$ DPUSCAN_DECLARATION_END

    QDb_Res1= Res1 ' globalem Zwischenspeicher entnehmen
End Function

Function QDb_Res2
'$$ DPUSCAN_DECLARATION_BEGIN
'$$ MODE Process
'$$ VAR QDb_Res2= %(I.Res2)
'$$ DPUSCAN_DECLARATION_END

    QDb_Res2= Res2 ' globalem Zwischenspeicher entnehmen
End Function
```

Alternativ kann die Funktion mehrere Rückgabewerte in einem Array zurückgeben. In DpuScan gibt es keine Arrays, deshalb verteilt das Script-Plugin die Array-Elemente auf DpuScan-Einzelvariablen gemäß der Zuordnung im Funktionskopf:

```
Function QDb_ResAll (par1, par2)
'$$ DPUSCAN_DECLARATION_BEGIN
'$$ MODE Process
'$$ VAR par1 = %(I.Par1)
'$$ VAR par2 = %(I.Par2)
'$$ VAR QDb_ResAll= %(I.Res0),%(I.Res1)

'$$ DPUSCAN_DECLARATION_END

Dim imResult(2)

    imResult(0)= par1+par2
    imResult(1)= par1*par2

    QDb_ResAll= imResult

End Function
```

## Typen von Rückgabewerten

DpuScan arbeitet ausschließlich mit Zeichenketten-Variablen. Rückgabewerte von Funktionen wandelt DpuScan entsprechend in Zeichenketten.

Der besondere Rückgabewert NULL ändert die zugeordnete DpuScan-Variable nicht.

### 4.2.4 Verfügbare Events (wie bei Broker)

DpuScan ruft exportierte und als Event-Handler gekennzeichnete Funktion bei folgenden Ereignissen auf:

```
Bei Taskstart (Task Start)
Bei Taskende (Task End)

Mit dem Anlegen eines Stapels (Batch Creation)

Vor dem Exportieren eines Stapels (Before Batch Export)
Nach dem Exportieren eines Stapels (After Batch Export)

Vor dem Finalisieren eines Stapels (Before Batch Finalization)
Nach dem Finalisieren eines Stapels (After Batch Finalization)

Zur Ereignisregel (As Event Rule)
```

Die folgenden Ereignisse treten auf: im Direct Mode sofort, imOpenJob Mode erst beim Finalisieren

```
Mit jeder Bildgruppe (Every Image Group)
Mit dem ersten Bild (First Image)
Mit dem letzten Bild (Last Image)
```

Mit dem Erzeugen einer Bilddatei (File Creation)

Mit dem Schließen einer Bilddatei (File Closing)

Mit dem Erzeugen eines Ordner (Path/Folder Creation)

Mit dem Schließen eines Ordner (Path/Folder Closing)

## 5 Konfiguration des Plugins

### 5.1 Selektion von Plugin und Subprofil

Das Plugin muss direkt an der Aufrufstelle konfiguriert werden, also in der Taskliste oder in der Applikationsdarstellung.

Für den Aufruf in der Taskliste stehen zwei Methoden zur Verfügung

- Plugin aufrufen
- Plugin aufrufen für jedes Bild

Dem Aufruf muss zugeordnet sein

- Script Plugin
- das Subprofil, in dem das an dieser Stelle zu rufende VBScript gespeichert ist

Für den Aufruf eines VBScripts von einer Schaltfläche stehen die Kommandos

- PluginAufrufParam01
- PluginAufrufParam02
- ...
- PluginAufrufParam32

für maximal 32 Schaltflächen zur Verfügung.

Innerhalb eines Makros können beliebig viele VBScripts aufgerufen werden. Verwendet werden hier wieder die Kommandos

- PluginAufrufParam01..32

jedoch ist jedem Kommando innerhalb des Makro unmittelbar ein VBScript zugeordnet. Der Index im Kommando ist bedeutungslos. Es gibt auch keine Überschneidungen mit den Schaltflächen-Konfigurationen gleichen Indexes.

Dem Kommando PluginAufrufParamXx werden zugeordnet

- Script Plugin
- das Subprofil, in dem das an dieser Stelle zu rufende VBScript gespeichert ist

## 5.2 Eingabe des VBScript Textes

Wo das Subprofil ausgewählt wird, kann über  in die Eingabe des Programmtextes verzweigt werden.

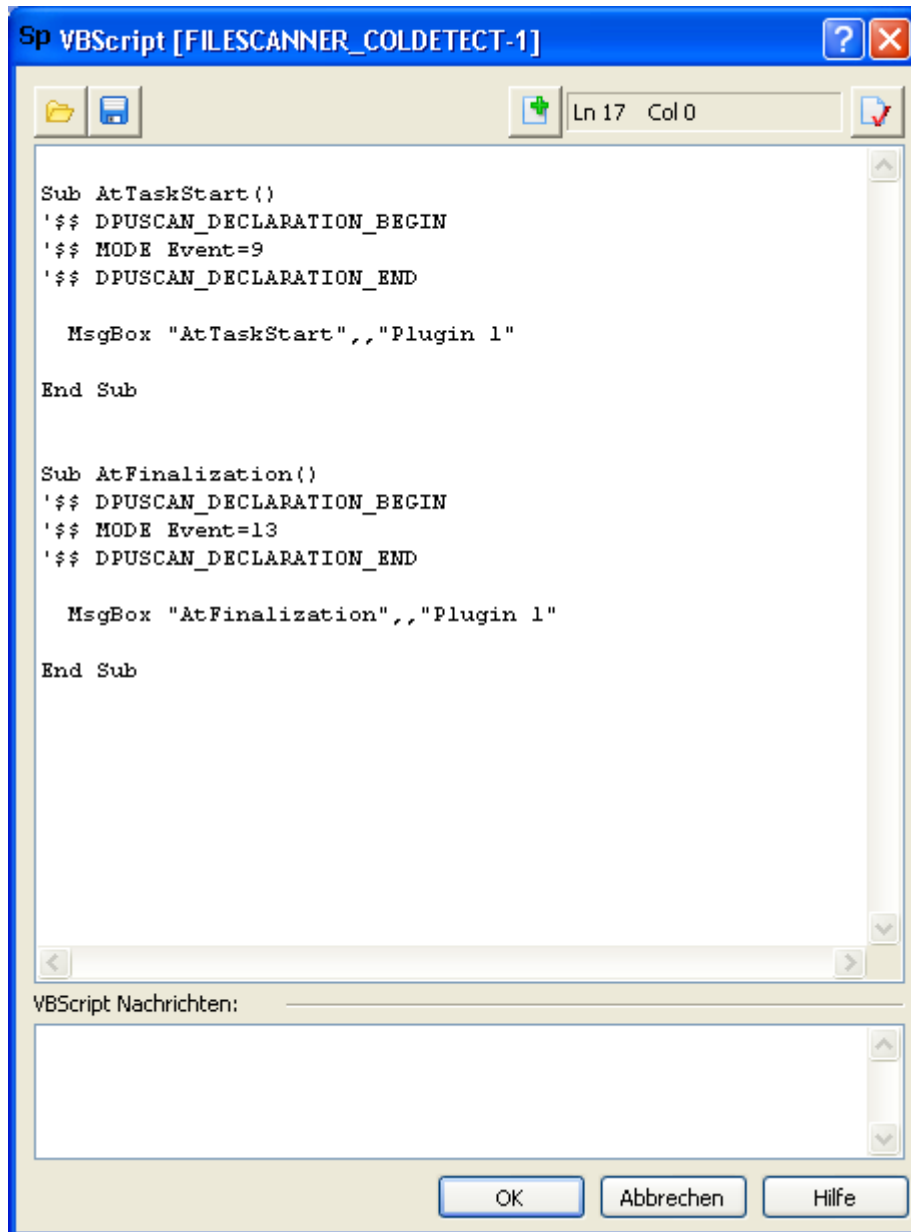




Abbildung 4 – Eingabe VBScript-Programmtext

Der angebotene Editor gestattet die Eingabe von VBScript-Text, das Laden solchen Textes von Datei und die Speicherung auf Datei. Die Datei ist jedoch kein Teil der Konfiguration, ist nicht an DpuScan gekoppelt. Es gilt einzig der VBScript-Text, der in dem Editfenster gespeichert.

Funktionen lassen sich über  hinzufügen. Als Funktionstyp stehen zur Auswahl die schon

erwähnten Process, Event, Selection.

Es ist möglich, über  einen Syntaxcheck durchzuführen.

### 5.3 Besondere Verfahren in VBScript

Für Datei-Ein- und Ausgaben steht unter VBScript das FileSystem-Objekt zur Verfügung

```
Dim Fso
Set Fso = CreateObject("Scripting.FileSystemObject")
```

Auf Datenbanken kann zugegriffen werden über

```
Dim Conn
Set Conn = CreateObject("ADODB.Connection")
```

Regular Expression lassen sich nutzen über

```
Dim Reg
Set Reg = New RegExp           ' Create a regular expression.
```

Wo HTTP-Transfer notwendig ist, hilft

```
Dim Http
Set Http = CreateObject("Msxml2.XMLHTTP.6.0")
```

XML-Verarbeitung ist möglich über

```
Dim XmlDoc
Set XmlDoc = CreateObject("Msxml2.DOMDocument.6.0")
```

Eine Dokumentation zu VBScript ist verfügbar bei MSDN

<http://msdn.microsoft.com>

nach einer Suche nach ‚VBScript‘.

Für die Nutzung der o.g. Objekte ADODB, XMLHTTP, DOMDocument stellt MSDN ebenfalls eine Referenz von Eigenschaften und Methoden bereit.



## 6 Laufzeitfehler, Debugging

Gibt es in einem DpuScan-Lauf Fehler in der Ausführung des VBScript , so signalisiert das VBScript den Fehler im DpuScan Warnungsfenster:



Der Konfigurationsdialog des Script Plugin kann den VBScript-Programmtext in eine Datei vom Typ VBS speichern. Windows kann eine solche Datei unmittelbar ausführen. Ein für Testzwecke zusätzlich in den Programmtext gebrachter Funktionsaufruf simuliert den Aufruf exportierter Funktionen. Dieser Aufruf muss von außerhalb jeglicher Funktion aus erfolgen. Der Aufruf sollte so gestaltet sein, dass er nur Stringparameter übergibt. Testausgaben können mit MsgBox auf dem Bildschirm gebracht werden.







Janich & Klass Computertechnik GmbH  
Zum Alten Zollhaus 24  
42281 Wuppertal  
Deutschland  
Tel.: +49 (0)202 2708-0  
Fax: +49 (0)202 700 625  
<http://www.janichklass.com>