Scripting

Mit Hilfe des Scripting Moduls kann speedyPDM ereignisorientiert angepasst werden.
Um das Scripting Modul nutzen zu können muss die Programmbibliothek modScript7.dll im speedy/Admin unter Anwendungsmodule hinzugefügt werden.
Sie haben die Wahl die Ereignisbehandlungsroutinen entweder in VBScript oder JScript zu schreiben.
Die Ereignisbehandlungsroutinen werden in einer Textdatei definiert. Das Scripting-Modul lädt die Script-Datei **„speedy.vbs“** bzw. **„speedy.js“** aus dem Unterverzeichnis **„.\script“**. Des weiteren werden aus dem selben Ordner noch die Script-Dateien **„document.*“**,**„container.*“**,**„swxSpeedy.*“**, **„aisSpeedy.*“** und **„seSpeedy.*“** aufgerufen.
Beispielcodes befinden sich im Verzeichnis **.\script\example**.

Script-Datei	Bemerkung
speedy.*	Hier werden allgemeine Funktionen des Speedy Explorers aufgerufen
document.*	Wird beim Neu-Dialog und beim Eigenschaften-Dialog des Dokuments aufgerufen
container.*	Wird beim Neu-Dialog und beim Eigenschaften-Dialog des Containers aufgerufen
file.*	Wird beim Eigenschaften-Dialog von Dateien aufgerufen
swxSpeedy.*	Wird bei Aktionen innerhalb von Solidworks aufgerufen
aisSpeedy.*	Wird bei Aktionen innerhalb von Autodesk Inventor aufgerufen
seSpeedy.*	Wird bei Aktionen innerhalb von SolidEdge aufgerufen

Ereignisroutinen

Die Ereignisroutinen haben, wenn nicht anders beschrieben, immer die gleichen Aufrufparameter.

```
Event_Function_Name(timing, handling, dictionary)
```

Parameter	Type	Description
timing	EventTimingEnum	Gibt den Zeitpunkt des Ereignisses an.
handling	HandlingCodeEnum	Rückgabewert, ob die Ereignisroutine das Ereignis behandelt hat und damit evtl. keine weitere Arbeit durch das System zu erfolgen hat.
dictionary	Dictionary - Object	Scripting.Dictionary aus der Scripting Runtime Library. Der Inhalt des Dictionaries kann variieren.

Konstanten

Der Parameter **timing** gibt an zu welcher Zeit das Ereignis ausgeführt wird.

```
' enum EventTimingEnum; Parameter := timing
Const kUpdateUI  = 0
Const kBefore    = 1
Const kAfter     = 2
Const kAbort     = 3
Const kFailed    = 4
```

Der Parameter **handling** gibt eine Information wie das Ereignis behandelt wurde zurück. Standart-Wert = 1

```
' enum HandlingCodeEnum; Parameter := handling
Const kEventHandled    = 0
Const kEventNotHandled = 1
Const kEventCanceled   = 2
```

Allgemeine Eigenschaften des Dictionary Objekts

Eigenschaft	Beschreibung
event.function	Name der Ereignisfunktion
event.timing	Zeitpunkt der Ereignisfunktion (siehe EventTimingEnum)
session.database	ADODB-Datenbankobjekt (ADODB.Connection) der Session Datenbank
session.user	Aktueller Speedy User
session.apptype	Applikationstyp: 1 := Service-Modul 4 := speedy-Admin 8 := speedy-Explorer 16 := speedy-Viewer 128 := Solidworks-Addin 256 := Inventor-Addin
vault.database	ADODB-Datenbankobjekt (ADODB.Connection) der Vault Datenbank
vault.name	Name des Vault-Objekts
obj_id	Objekt-Id als Zeichenkette in der Form „{0-1-2-3-4-5}„
obj_id.id	Die eigentliche Objekt-Id
obj_id.type	Der Objekttyp
obj_name	Name des Objekts. Bei Ordner wird hier der Ordnername angegeben.
obj_desc	Beschreibung des Objekts. Bei Ordner wird hier die Ordnerbeschreibung angegeben.
obj_description	
obj_class	Klassenname des Objekts (z.B. CZvContainer für Ordner).

Die beiden Eigenschaften session.database und vault.database liefern kein Ergebnis, wenn mit dem Datenbank-Direkttreiber „MYSQL“ gearbeitet wird.

Globale Funktionen

Application - Objekt

In den Scriptfunkktionen steht das globale Object ***Application*** mit folgenden Funktionen zur Verfügung:

Application.GetSessionDb

Gibt das ADODB.Connection Objekt der Session Datenbank zurück.

Diese Funktion liefert kein Ergebnis, wenn mit dem Datenbank-Direkttreiber „MYSQL“ gearbeitet wird.

Application.GetRootPath

Gibt den Root Path, wie in der dwKernel.ini eingestellt, zurück.

Application.GetVaultCount

Gibt die Anzahl der Vault datenbank zurück.

Application.GetVaultDbAt(iIndex)

Gibt das ADODB.Connection Object der entsprechenden Vault Datenbank zurück.

Application.GetVaultNameAt(iIndex)

Gibt den Namen der Vault Datenbank zurück.

Application.GetTempFileName

Gibt einen temporären Dateinamen im aktuellen Temp-Verzeichnis zurück.

Application.GetTempPath

Gibt das aktuelle Temp-Verzeichnis zurück.

Application.GetAppPath

Gibt das Installationsverzeichnis der Anwendung zurück.

Application.ProgressStart(lMin, lMax, sTitle, sMessage)

Beginnt einen Dialog mit Fortschrittsanzeige.

Application.ProgressStop

Stoppt die Fortschrittsanzeige.

Application.ProgressText(sText)

Setzt den Text in der Fortschrittsanzeige.

Application.ProgressStep

Führt einen Schritt in der Fortschrittsanzeige aus.

Application.Log(lSeverity, sMessage)

Protokolliert einen Text in der aktuellen Log-Datei.

Application.InvokeCommand(sCallBack, wParam, lParam)

Ruft einen speedy Zusatzbefehl auf.

Application.HasCommand(sCallBack)

Überprüft, ob der Zusatzbefehl vorhanden ist.

Application.ConsoleExecute(sCmd, sDirectory, lWait)

Führt ein Programm/Aufruf auf der Konsole aus.

Parameter	Beschreibung
sCmd	Befehlszeile
sDirectory	Verzeichnis in dem der Befehl ausgeführt werden soll.
lWait	Soll auf das Ende der Befehlsausführung gewartet werden.

Application.DlgLookup(ByRef Result, DataName, Connection, DataSource, InitialValue, Option)

Öffnet einen Lookup-Dialog zur Auswahl.
Die Funktion gibt True bei erfolgreicher Auswahl zurück und False bei Abbruch durch den User.

Parameter	Beschreibung
Result[String]	Ergebnis der Auswahl, Wert der 1. Spalte der gewählten Zeile in der Auswahltabelle.
DataName[String]	a) Name des Eigenschaftenformats. Im Eigenschaftenformat wird die Datenbankverbindung und das SQL Statement definiert. Die Werte für Connection und DataSource bleiben leer. b) Name der Datenbankverbindung. Bei externer Datenbank optional. z.B. „session.database“ für eigene Datenbank.
Connection[String]	ConnectionString zum Aufbau der Datenbankverbindung.
DataSource[String]	SQL Statement zur Ermittlung der Auswahltabelle. Die 1. Spalte ergibt das Ergebnis des Lookup.
InitialValue[String]	Vorab gewählter Wert.
Option[long]	Wird zur zeit nicht unterstützt.

Application.EditObject(sItemId)

Bearbeitet das durch ItemId definierte Element mit dessen nativem Eigenschaften-Dialog.

Application.EditProperties(sItemId, Properties, bShowInvisible, sTitle, sDescription, sProfile)

Bearbeitet das durch ItemId definierte Element in einem vereinfachten Eigenschaften-Dialog.

Parameter	Beschreibung
sItemId (String)	Id des zu bearbeitenden Elements ({x-x-y-y-z-z})
Properties (String-Array)	Array mit Eigenschaftennamen, die im Dialog bearbeitet werden sollen. Für die Eigenschaften muss es jeweils ein definiertes Eigenschaftenformat geben.
bShowInvisible (Boolean)	Sollen auch auch unsichtbar (Größe < 0) definierte Eiegnschaften im Dialog darstellen.

Parameter	Beschreibung
sTitle (String)	Titelzeile der Dialogüberschrift.
sDescription (String)	Beschreibungszeile der Dialogüberschrift.
sProfile (String)	Profilname unter dem die Fenstergröße in der Registry gespeichert wird.

Application.GetDocumentPath(DocNo)

Gibt den Ablagepfad des Dokuments zurück.

```
doc_path = Application.GetDocumentPath(dictionary.item("dm_docno"))
```

Application.DbExecute(Connection, Statement)

Führt das SQL-Statement auf der benannten Datenbankverbindung aus.
Wird keine Verbindung (Connection) angegeben, wird die Datenbank der Session verwendet.
In Multi-Vault-Umgebungen kann der Name des Vault angegeben werden, um die Datenbank des Vault zu verwenden.
Die Funktion gibt 0 zurück, wenn kein Fehler vorliegt.
Die Funktion liefert kein Ergebnis aus dem SQL-Statement zurück.

```
Call DbExecute("", "UPDATE dm_document d inner join dm_version v ON
d.doc_did=v.ver_did AND d.doc_rev=v.ver_major AND d.doc_ver=v.ver_minor
inner join dm_prop_cad p on v.ver_vid=p.prop_did SET p.NORM = 'DIN 931'
WHERE d.doc_docno = 'BS01516';")
```

Application.DbQuery(ByRef Result, Connection, Statement)

Führt das SQL-Abfrage auf der benannten Datenbankverbindung aus.
Wird keine Verbindung (Connection) angegeben, wird die Datenbank der Session verwendet.
Rückgabewert: 0:=Kein Fehler, Sonst Fehler-Code

In Multi-Vault-Umgebungen kann der Name des Vault angegeben werden, um die Datenbank des Vault zu verwenden.
Die Funktion liefert die Werte der in der Abfrage definierten Felder in einem Array zurück.
Beispiel:

```
Call DbQuery(result, "", "SELECT lst_idx,lst_name,lst_desc FROM lng_lst
WHERE lst_typ='PROJEKTSTATUS';")

Ergebnis:
0 0 "Erstellt"
1 1 "In Arbeit"
```

```
2 2 "Angehalten"
3 3 "Beim Kunden"
4 4 "Im Bau"
5 5 "Fertig"
```

Beispiel:

```
Dim sql, result
sql = "SELECT lst_name,lst_desc FROM lng_lst WHERE lst_typ='bomtype'"
result = Null
if Application.DbQuery(result, "", sql) = 0 And not IsNull(result) Then
   MsgBox result(0, 0)
   MsgBox result(0, 1)
end if
```

Application.DbLookup(ByRef Result, Connection, Expression, Domain, Criteria)

Mit der DbLookup-Funktion können Sie den Wert eines bestimmten Feldes (Expression) aus einer bestimmten Datensatzgruppe (einer Domäne) abrufen.
Mit der Angabe von Criteria definieren Sie ein Suchkriterium.
Wird keine Verbindung (Connection) angegeben, wird die Datenbank der Session verwendet.
Rückgabewert: 0:=Kein Fehler, Sonst Fehler-Code

In de Praxis werden die 3 Werte zu einem SQL-Statement zusammengesetzt und der erste gefundene Wert ist das Ergebnis der Suche:

```
SELECT <Expression> FROM <Domain> WHERE Criteria
```

```
Call DbLookup(result, "", "MAX(doc_docno)", "dm_document", "") - Liefert die
größte Dokumentnummer
Call DbLookup(result, "", "sett_value", "dm_setting",
"sett_key='bom.autosorted'")  - Liefert den Einstellungswert des
Konfigurationsparameters [bom.autosorted]
```

Application.Sleep(Milliseconds)

Hält die Ausführung der Anwendung an, bis das Timeout-Intervall abgelaufen ist.

```
Call Application.Sleep(1000)
```

Application.GetSetting(Key)

Gibt den Wert einer Einstellung zurück.

```
archive_path = Application.GetSetting("session.path.archive")
```

Application.SetSetting(Key, Value)

Setzt den Wert einer Einstellung.

```
Call Application.SetSetting("document.new.docno.zaehler", 4711)
```

Application.GetDocumentProperty(DocNo, Property)

Gibt eine Dokumenteigenschaft des Dokuments zurück.
Mit der Methode GetDocumentProperty können alle Dokument-Eigenschaften oder auch Struktur-Eigenschaften abgefragt werden besonders dann wenn diese nicht im dictionary enthalten sind.

```
s = Application.GetDocumentProperty(dictionary.item("dm_docno"),
“projectcontainer.PROJECTMANAGER“)
```

Clipboard - Objekt

Globales Objekt mit Funktionen zur Zwischenablage.

Clipboard.Clear

Leert die Zwischenablage

Clipboard.Copy(Text)

Kopiert den Text in die Zwischenablage

Clipboard.Paste

Gibt den aktuellen Text in der Zwischenablage zurück.

```
Text = Clipboard.Paste()
```

speedy.*

Allgemeine Objekt-Ereignisse

In der Ereignisroutine werden außer den allgemeinen Eigenschaften der Objekte auch noch objektspezifische Eigenschaften mit ins Dictionary geschrieben.

Object_onEvent(timing, handling, dictionary)

Ein Ereignis passiert. Der Ereignisname wird im Dictionary aufgeführt.

Eigenschaft	Beschreibung
event	Übergibt einen Ereignis-Namen
wparam	
lparam	

Folgende allgemeine Ereignisse sind definiert und fürs Skripting geeignet:

Name	Beschreibung
Command::ExportDocument	
Command::ExportBomTable	
CDwPropertyEdit::OnSearch	
CDwPropertyEdit::OnEdit	
SWXAddin::ActiveModelDocChangeNotify	

Object_onCreate(timing, handling, dictionary)

Ein Objekt (z.B. Ordner) wird in der Datenbank erstellt. timing = after: Objekteigenschaften des neuen Objekts werden mit ins Dictionary geschrieben

Object_onDelete(timing, handling, dictionary)

Ein Objekt wird in der Datenbank gelöscht. timing = before: Objekteigenschaften des zu Löschenden Objekts werden mit ins Dictionary geschrieben

Object_onInit(timing, handling, dictionary)

Ein Object-Container wird initalisiert.

Object_onUpdate(timing, handling, dictionary)

Ein Objekt wird aktualsiert, d.h. Änderungen am Objekt werden in die Datenbank geschrieben.

Object_onCopy(timing, handling, dictSource, dictTarget, dictionary)

Ein Object wird kopiert.

Parameter	Beschreibung
dictSource	Eigenschaften des Quell-Objekts
dictTarget	Eigenschaften ds Ziel-Objekts
dictSource	Allgemeine Eigenschaften

Object_onCreateObject(timing, handling, dictionary)

Ein neues Objekt wird mit Benutzerinteraktion (Neu-Dialog) erstellt.

Eigenschaft	Beschreibung
CreateObject.flag	
CreateObject.hwnd	
CreateObject.subitem	
CreateObject.param	
CreateObject.file	
CreateObject.usetemplate	
CreateObject.useextension	
CreateObject.deleteorigin	
CreateObject.saveascopy	

Object_onEditObject(timing, handling, dictionary)

Die Eigenschaften eines Objekts sollen mit Benutzerinteraktion bearbeitet werden.

Eigenschaft	Beschreibung
EditObject.flag	
EditObject.hwnd	
EditObject.subitem	
EditObject.param	

Object_onBuildDocName(docno, dictVault, dictParent, dictOrigin, dictDoctype, dictProperties, timing, handling)

Eine neue Dokumentnummer soll genriert werden.

Parameter	Beschreibung
docno	Rückgabe der neuen Dokumentnummer
dictVault	Eigenschaften des aktuellen Vaults
dictParent	Eigenschaften des Elternelements (z.B. Ordner)
dictOrigin	Eigenschaften des Ursprung Objekts (z.B. Dokument beim Kopieren)
dictDoctype	Eigenschaften des Ordnertyps
dictProperties	Sonstiges Eigenschaften

Dokument-Ereignisse

In der Ereignisroutine werden außer den Allgemeinen Eigenschaft der Dokumente auch noch Dokumentspezifische Eigenschaften mit in das Dictionary geschrieben.

Document_onEvent(timing, handling, dictionary)

Eigenschaft	Beschreibung
„event“	Übergibt einen Ereignis-Namen

Document_onBeforeImport(timing, handling, dictionary)

Eigenschaft	Beschreibung
timing	Immer kBefore
„filename“	Übergibt den Datei-Namen

Document_onCreate(timing, handling, dictionary)

Ereignis beim Erstellen eines Dokuments.

Parameter	Beschreibung
timing = kAfter	Dokumenteigenschaften des neuen Dokuments werden mit ins Dictionary geschrieben.
handling	Wenn bei timing=kAfter Eigenschaften direkt in der Datenbank angepasst werden muss das handling=kEventHandled gesetzt werden damit Die Dokument-Eigenschaften vor dem weiter arbeiten erneut eingelesen werden.

Beispiel-Code setzt die Eigenschaft ARTIKEL_ANLAGE=1 bei Dokumenten vom Typ „E“ oder „K“:

```
Function Document_onCreate(timing, handling, dictionary)
    Dim dm_docno
    Dim dm_doctype
    Dim sql

    if timing = kAfter Then
        dm_doctype = dictionary.item("dm_doctype")
        dm_docno = dictionary.item("dm_docno")
        if dm_doctype="E" Or dm_doctype="K" Then
            sql = "UPDATE dm_document d inner join dm_version v on
d.doc_did=v.ver_did and d.doc_rev=v.ver_major and d.doc_ver=v.ver_minor
inner join dm_prop_cad p on v.ver_vid=p.prop_did SET p.ARTIKEL_ANLAGE = 1
WHERE d.doc_docno='" & dm_docno & "'"
            Application.DbExecute "", sql
            handling = kEventHandled
            Document_onCreate = true
        end if
    end if
End Function
```

Document_onDelete(timing, handling, dictionary)

timing = before: Dokumenteigenschaften des zu Löschenden Dokuments werden mit ins Dictionary geschrieben

Eigenschaft	Beschreibung
„filename“	Übergibt den Datei-Namen

Document_onCreateVersion(timing, handling, dictionary)

timing = after: Dokumenteigenschaften der neuen Version des Dokuments werden mit ins Dictionary geschrieben

Eigenschaft	Beschreibung
„revisionno“	Erstellt die neue Freigabenummer
„major“	Schaut nach höherer Freigabenummer
„minor“	Schaut nach niedriger Freigabenummer
„status“	Übergibt den aktuellen Status

Document_onDeleteVersion(timing, handling, dictionary)

timing = before: Dokumenteigenschaften der zu löschenden Version des Douments werden mit ins

Dictionary geschrieben

Eigenschaft	Beschreibung
„major“	Schaut nach höherer Freigabenummer
„minor“	Schaut nach niedriger Freigabenummer

Document_onRemoveFile(timing, handling, dictionary)

timing = before: Dokumenteigenschaften des zu entfernenden Douments werden mit ins Dictionary geschrieben

Eigenschaft	Beschreibung
„filename“	Übergibt den Datei-Namen

Document_onStatusChange(timing, handling, dictionary)

timing = before: Dokumenteigenschaften des alten Dokuments werden mit ins Dictionary geschrieben
timing = after: Dokumenteigenschaften des neuen Dokuments werden mit ins Dictionary geschrieben

Eigenschaft	Beschreibung
„status“	Übergibt den Status

Document_onRelease(timing, handling, dictionary)

Eigenschaft	Beschreibung
„comment“	Übergibt ein Kommentar
„major“	Letzte Freigabe
„deletelocal“	Löscht die lokale Freigabe

Document_onPendingApproval(timing, handling, dictionary)

Eigenschaft	Beschreibung
comment	Übergibt ein Kommentar

Document_onApprove(timing, handling, dictionary)

Eigenschaft	Beschreibung
„comment“	Übergibt ein Kommentar

Document_onUndoRelease(timing, handling, dictionary)

Eigenschaft	Beschreibung
„comment“	Übergibt ein Kommentar

Document_onCheckIn(timing, handling, dictionary)

Eigenschaft	Beschreibung
„extension“	String. Dateierweiterung, die eingecheckt werden soll.
„deletelocal“	Boolean. Soll die lokale Datei beim einchecken gelöscht werden.
„force“	Boolean. Erzwingt das einchecken.

Document_onCheckInNew(timing, handling, dictionary)

Eigenschaft	Beschreibung
„file“	Übergibt eine Datei
„origin“	Übergibt den Ursprungsdateiname
„useextension“	Nutzt die Datei-Extension separat
„deletelocal“	Löscht die lokale Freigabe

Document_onCreateFile(timing, handling, dictionary)

timing = after: Dokumenteigenschaften des neuen Douments werden mit ins Dictionary geschrieben

Eigenschaft	Beschreibung
„template“	Übergibt das Template separat
„extension“	Übergibt die Datei-Extension separat

Document_onCheckOut(timing, handling, dictionary)

Eigenschaft	Beschreibung
„extension“	String. Dateierweiterung, die ausgecheckt werden soll.
„force“	Boolean. Erzwingt das auschecken.
„path“	String. Verzeichnis in das ausgecheckt wird.

Document_onCheckOutReleased(timing, handling, dictionary)

Eigenschaft	Beschreibung
„extension“	String. Dateierweiterung, die ausgecheckt werden soll.
„force“	Boolean. Erzwingt das auschecken.

Eigenschaft	Beschreibung
„path“	String. Verzeichnis in das ausgecheckt wird.

Document_onCheckOutIn(timing, handling, dictionary)

Eigenschaft	Beschreibung
„extension“	String. Dateierweiterung, die ausgecheckt werden soll.
„force“	Boolean. Erzwingt das auschecken.

Document_onUndoCheckOut(timing, handling, dictionary)

Document_onUndoCheckOutReleased(timing, handling, dictionary)

Document_onCopyOut(timing, handling, dictionary)

Document_onOpen(timing, handling, dictionary)

timing = after: Dokumenteigenschaften des zu öffnenden Dokuments werden mit ins Dictionary geschrieben

Eigenschaft	Beschreibung
„extension“	Übergibt die Datei-Extension separat
„readonly“	Öffnet die Datei nur lesend
„waitforexecute“	Wartet bis etwas ausgeführt ist

Document_onPrint(timing, handling, dictionary)

timing = after: Dokumenteigenschaften des zu Druckenden Dokuments werden mit ins Dictionary geschrieben

Eigenschaft	Beschreibung
„extension“	Übergibt die Datei-Extension separat

Document_onView(timing, handling, dictionary)

timing = after: Dokumenteigenschaften des zum Viewen geöffneten Dokuments werden mit ins Dictionary geschrieben

Eigenschaft	Beschreibung
„extension“	Übergibt die Datei-Extension separat

Document_getRevisionLabel(timing, handling, dictionary)

Das Revisions-Label kann angepasst werden. Übergabe ist das Standard Label in der Form “<major>.<minor>“.
Wenn die Funktion das Label verändert hat muss sie true zurück liefern und den Handling-Code setzen.
Im Dictionary ist die Eigenschaft „dm_revlabel“ entsprechend anzupassen.
Der Konfigurationsparameter [session.revlabel] ist auf 0 zu setzen.

Eigenschaft	Beschreibung
„dm_revlabel“	Revision-Label

```
Public Function Document_getRevisionLabel(timing, handling, dictionary)
  Dim revlabel
  Dim major
  Dim minor
  Dim i1

  revlabel = dictionary.Item("dm_revlabel")
  i1 = InStr(revlabel, ".")
  If i1 > 0 Then
    major = CLng(Left(revlabel, i1))
    minor = CLng(Mid(revlabel, i1+1))
  Else
    major = 0
    minor = 0
  End If

  if minor = 0 then
    Select Case CLng(major)
    Case 0
    dictionary.Item("dm_revlabel") = "*"
    Case 1
    dictionary.Item("dm_revlabel") = "-"
    Case Else
    dictionary.Item("dm_revlabel") = CStr(major-1)
    end select
    handling = kEventHandled
    Document_getRevisionLabel = true
  end if
```

```
End Function
```

Document Operations-Ereignisse

Ereignisse die sich beim Kopieren von Dokumenten ergeben.

DocumentOperations_onCopyDocument

```
Function DocumentOperations_onCopyDocument(timing, handling, source, target,
dictionary)
```

Eigenschaft	Beschreibung
source	Eigenschaften des Quell Dokuments.
target	Eigenschaften des Ziel Dokuments.
dictionary	Allgemeine Eigenschaften und evtl. Rückgabewert.

Es kann sowohl vor der Kopier-Operation als auch nach dem Kopieren reagiert werden (timing).
Wenn bei einem kopierten Dokument Daten geändert werden sollen (timing=kAfter) muss dies direkt auf der Datenbank erfolgen. Damit diese Änderung vom System erkannt wird müssen folgende Rückgaben erfolgen:

```
handling = kEventHandled
DocumentOperations_onCopyDocument = 0
dictionary.item("target.refresh") = True
```

Das nachfolgende Beispiel ändert den „LAGERORT“ in dem kopierten Dokument und vergibt eine neue Nummer.
In der Tabelle [dm_setting] muss zuvor ein Eintrag für „document.new.docno.zaehler“ erstellt werden.

```
Function DocumentOperations_onCopyDocument(timing, handling, source, target,
dictionary)
    On Error Resume Next
    Dim docno
    Dim doctype
    Dim dtype_proptable
    Dim zaehler
    Dim sql
    if timing = kAfter then
        docno = target.item("dm_docno")
        doctype = target.item("dm_doctype")
        'MsgBox "DocumentOperations_onCopyDocument:=" & docno
        if DbLookup(dtype_proptable, "", "dtype_proptable", "dm_doctype",
"dtype_name='" & doctype & "'") = 0 then
            if CStr(dtype_proptable) = "dm_prop_cad" then
                if DbLookup(zaehler, "", "sett_value", "dm_setting",
"sett_key='document.new.docno.zaehler'") = 0 then
```

```
                    zaehler = CLng(zaehler) + 1

                    sql = "UPDATE dm_setting SET sett_value ='" &
Cstr(zaehler) & "' where sett_key ='document.new.docno.zaehler'"
                    'MsgBox sql
                    Call DbExecute("", sql)
                    sql = "UPDATE dm_document d " & _
                        "inner join dm_version v on d.doc_did=v.ver_did and
d.doc_rev=v.ver_major and d.doc_ver=v.ver_minor " & _
                        "inner join dm_prop_cad p on v.ver_vid=p.prop_did "
& _
                        "set p.LAGERORT='" & Cstr(zaehler) & "' " & _
                        "WHERE d.doc_docno='" & docno & "'"
                    'MsgBox sql
                    Call DbExecute("", sql)
                    handling = kEventHandled
                    DocumentOperations_onCopyDocument = 0
                    dictionary.item("target.refresh") = True
                end if
            end if
        end if
    end if
End Function
```

Container-Ereignisse

Container_onDeleteBomStructure(timing, handling, dictionary)

Eigenschaft	Beschreibung
„docno“	Übergibt die Dokumenten-Nummer

Container_onAddDocument(timing, handling, dictionary)

Eigenschaft	Beschreibung
Document	Alle Eigenschaften des Dokument werden übergeben
Container	Alle Eigenschaften des Containers werden übergeben

Container_onRemoveDocument(timing, handling, dictionary)

Eigenschaft	Beschreibung
Document	Alle Eigenschaften des Dokument werden übergeben
Container	Alle Eigenschaften des Containers werden übergeben

Nummern-Generator-Ereignisse

ParamName_onBuild(timing, handling, dictionary)

Erstellt eine neue Nummer für ein neues Dokument, eine neue Variante, einen neuen Ordner.
Die Funktion wird immer dann aufgerufen, wenn ein neuer Name/neue Nummer basierend auf der Nummerngenerator-Syntax erstellt werden muss.

Eigenschaft	I/O	Beschreibung
	Alle Eigenschaften der Eltern Objekte, um eine neue Nummer zu erstellen.	
„docno“	input/output	Bereits berechnete Dokumentnummer. Rückgabe der berechneten Dokumentnummer.
„class“	input	„CDwDocumentName“ := Nummerngenerator für ein neues Dokument. „CZvParamName“ := Nummerngenerator für alles andere (Ordner, Variante, ...)

Beispiel Dokumentnummer generieren:

```
Function ParamName_onBuild(ByVal timing, handling, dictionary)
    On Error Resume Next
    Dim conn ' As ADODB.Connection
    Dim rs ' As ADODB.Recordset
    Dim fld ' As ADODB.Field
    Dim sql
    Dim myYear
    Dim myMonth
    Dim myDocno
    Dim myMaxnumber
    Dim doctypkennung
    Dim doctype

    If timing = kBefore Then
        If dictionary.item("class") = "CDwDocumentName" Then
            'MsgBox "dm_doctype:=" & dictionary.item("dm_doctype")
            'Berechnen der Dokumentennummer
            myYear = DatePart("yyyy",Now())
            'MsgBox myYear
            myDocno = 0
            myMaxnumber = 0
            Set conn = dictionary.Item("session.database")
            Set rs = CreateObject("ADODB.Recordset")
            sql = "SELECT MAX(RIGHT(d.doc_docno,6)) FROM dm_document d WHERE
d.doc_docno REGEXP '[-]{1}[1]{1}[0-9]*$'"
            rs.Open sql, conn, adOpenForwardOnly, adLockReadOnly
            If rs.State = adStateOpen and Not rs.EOF Then
                for each fld in rs.Fields
                    myMaxnumber = fld.Value
```

```
                next
            End if

            If IsNull(myMaxnumber) Then
                myMaxnumber = 0
            End if
            myMaxnumber = CLng(myMaxnumber) + 1
            Select case dictionary.item("dm_doctype")
                case "A"
                 doctypkennung = "00-1"
                case "B"
                 doctypkennung = "00-1"
                case "E"
                 doctypkennung = "01-1"
                case "ES"
                 doctypkennung = "05-1"
                case "GRT"
                 doctypkennung = "06-1"
                case "HT"
                 doctypkennung = "99-1"
                case "K"
                 doctypkennung = "02-1"
                case "N"
                 doctypkennung = "03-1"
                case "NE"
                 doctypkennung = "04-1"
                case "S"
                 doctypkennung = "08-1"
                case "VB"
                 doctypkennung = "11-1"
                case "VS"
                 doctypkennung = "07-1"
                case else
                 doctypkennung = "00-1"
            End select
            myDocno = Right(myYear,2) + "-" + doctypkennung + Right("000000"
& CLng(myMaxnumber), 6)
            'MsgBox myDocno
            doctype = dictionary.item("dm_doctype")
            'Setzen von Dokumentennummer
            dictionary.Removeall
            dictionary.add "docno", myDocno
            handling = kEventHandled
            ParamName_onBuild = true

        End if
    End if
End Function
```

ParamName_onGetClass(timing, handling, dictionary)

Eigenschaft	Beschreibung
	Alle Eigenschaften des Dokument werden übergeben
„class“	Ermittelter Klassenname als Rückgabe.

File-Ereignisse

Ereignisse, die im Zusammenhang mit Dateioperationen auftreten.

FileOperations_onReReferenceFiles(timing, handling, params, mapfiles)

Parameter	Beschreibung
params	Dictionary mit allgemeinen Eigenschaften
mapfiles	Dictionary mit Datei-Mapping Alter-Dateiname⇔Neuer-Dateiname

Params-Eigenschaft	Beschreibung
source	Übergibt den Datei-Pfad
target	Übergibt den Zielpfad
msg	Sollen Fehlermeldungen ausgegeben werden
force	Erzwingen

FileOperations_onGetFileDependencies(timing, handling, dictionary, dependencies)

Parameter	Beschreibung
dictionary	Allgemeine Eigenschaften
dependencies	String-Array mit den Dependencies

FileOperations_onReplaceReferencedFiles(timing, handling, dictionary, filename, mapReplacements)

Parameter	Beschreibung
dictionary	Allgemeine Eigenschaften
filename	
mapReplacements	

FileOperations_onFileSaveAs(timing, handling, dictionary)

Ereignis-Routine wird von der Zusatzfunktion „Speichern als“ im Explorer aufgerufen.
Damit kann bei den von „Speichern als“ unterstützten Dateiformaten gesondert reagiert werden.

Wird mit einem Skript auf ein Dateiformat reagiert ist darauf zu achten, dass das Scripting-Modul (modScript7.dll) vor den CAD-Modulen (mod<CAD>.dll) geladen wird und bei erfolgreicher Abarbeitung den Übergabeparameter handling mit kEventHandled zurückliefert.

Eigenschaft	Beschreibung
filename	Übergibt den Datei-Pfad
target	Übergibt den Zielpfad
pages	Übergibt ob alle Seiten gespeichert wird oder nur die erste

FileOperations_onAppRelease()

FileOperations_onFileExport(timing, handling, dictionary)

Ereignis-Routine wird beim Exportieren einer Datei aufgerufen.

Dictionary-Eigenschaft	Beschreibung
source	Dateiname inkl. Pfad der Quelldatei.
target	Dateiname inkl. Pfad der Zieldatei.

Das Dictionary enthält die Eigenschaften des Dokuments sowie die Eigenschaften des Dateiobjekts. Die internen Eigenschaften Dateiobjekts beginnen mit „file_“. Die zusätzlichen Eigenschaften des Dateiobjekts beginnen mit „file.“ damit keine Überschneidung mit den Zusatzeigenschaften des Dokuments bestehen.
Wenn das timing-Argument den Wert kUpdateUI hat kann der Dictionary Eintrag für den „target“ durch einen neuen Dateinamen inkl. Pfadangabe ersetzt werden. Wurde das Exportziel „target“ verändert muss das Argument handling den Wert kEventHandled erhalten.
z.B.:

```
Public Function FileOperations_onFileExport(ByVal timing, handling,
dictionary)
   Dim old_target, target, BEN1, ext
   if timing = kUpdateUI then
      old_target = dictionary.Item("target")
      ext = PathFindExtension(old_target)
      target = PathRemoveFileSpec(old_target)
      BEN1 = dictionary.Item("BEN1")
      BEN1 = Replace(BEN1, " ", "_")
      target = PathCombine(target, BEN1)
      target = PathAddExtension(target, ext)
      dictionary.Item("target") = target
      handling = kEventHandled
   end if
End Function
```

Damit die Ereignis-Routine auch beim Drag&Drop aufgerufen wird muss der

Konfigurationsparameter [session.export.hasfileexportevent] aktiviert sein.

Import-/Export-Ereignisse

IO_onImport(timing, handling, import, objects, file)

Parameter	Beschreibung
import	Allgemeine Eigenschaften sowie die Eigenschaften der Importbeschreibung
objects	Array von Objekt-Eigenschaften
file	Dateiname der Importdatei.

IO_onImportFile(timing, handling, import, objects, file)

Parameter	Beschreibung
import	Allgemeine Eigenschaften sowie die Eigenschaften der Importbeschreibung
objects	Array von Objekt-Eigenschaften
file	Dateiname der Importdatei.

IO_onImportObject(timing, handling, import, dictionary, first)

Parameter	Beschreibung
import	Allgemeine Eigenschaften sowie die Eigenschaften der Importbeschreibung
dictionary	Eigenschaften des zu importierenden Objekts
first	Boolean. Erstes Objekt des laufenden Imports.

IO_onExport(timing, handling, dictionary, objects, filename)

Paramater	Beschreibung
timing	
handling	
dictionary	Allgemeine Eigenschaften der Session und des Export-Objekts.
objects	Collection von Dictionary Objekten mit jeweils den Eigenschaften der zu exportierenden Objekten.
filename	Dateiname der Exportdatei, wenn in eine Datei exportiert wird.

Makro Befehle

OnCommand01 - OnCommand99

Funktion, die per kundenspezifischer Anpassung der Ribbonleiste, aufgerufen werden kann.

Parameter	Beschreibung
dictionary	Dictionary mit allgemeinen Eigenschaften der Session und des Vaults.
documents	Array mit allen beim Makroaufruf markierten Dokumenten (Dokumentnummer).

Erweiterung der Ribbonleiste um ein Schaltflächenelement:

```
   <ELEMENT>
    <ELEMENT_NAME>Button</ELEMENT_NAME>
    <ID>
       <NAME>OnCmdScript01</NAME>
       <VALUE>24500</VALUE>
    </ID>
    <TEXT>Befehl 01</TEXT>
    <INDEX_SMALL>8</INDEX_SMALL>
    <INDEX_LARGE>8</INDEX_LARGE>
   </ELEMENT>
```

Ribbonbefehl	Befehls-ID	Scriptbefehl
OnCmdScript01	24500	OnCommand01
...	...	...
OnCmdScript99	24599	OnCommand99

Beispiel Script-Funktion:

```
Function OnCommand01(dictionary, documents)
   On Error Resume Next
   Dim i, count

   MsgBox "OnCommand01: " & dictionary.Item("session.user")
   count = UBound(documents) - LBound(documents)
   For i=0 To count
      MsgBox "document[" & i & "]:=[" & documents(i) & "]"
   Next
End Function
```

document.*

Die Script-Datei „document“ enthält Ereignisse, die im Neu- und Eigenschaften-Dialog eines Dokuments aufgerufen werden. Sobald die Script-Datei im Script-Ordner gefunden wurde, erscheint neben der Dokumentnummer eine Schaltfläche.
Beispiel-Codes befinden sich im Verzeichnis **../script/example/document.example.vbs**.

OnBrowse (kBefore, dictionary)

Wird durch betätigen der Schaltfläche rechts neben der Dokumentnummer aufgerufen

Parameter	Beschreibung
kBefore	Zeitpunkt wann die Aktion ausgeführt werden soll (hier unrelevant)
dictionary	Übergibt das Dictionary Objekt

Rückgabe: true oder 1 := Das Dictionary enthält Eigenschaften, die in den Dialog zu übernehmen sind.

```
Function OnBrowse(lEventTiming, dictionary)
  OnBrowse = False
  If dictionary.Item("dlgmode") = "new" Then
    ' z.B. Dictionary mit Eigenschaften befüllen, die im Neu-Dialog zu
übernehmen sind
    OnBrowse = True
  ElseIf dictionary.Item("dlgmode") = "edit" Then
    ' z.B. Dictionary mit Eigenschaften befüllen, die im Eigenschaften-
Dialog zu übernehmen sind
    OnBrowse = True
  End If
End Function
```

OnValidate(kBefore, dictionary)

Wird beim Bestätigen des Neu-Dialogs vor dem Erstellen eines neuen Dokuments aufgerufen

Parameter	Beschreibung
kBefore	Zeitpunkt wann die Aktion ausgeführt werden soll (hier unrelevant)
dictionary	Übergibt das Dictionary Objekt

Rückgabe: 1 := Das Dictionary enthält Eigenschaften, die in den Dialog zu übernehmen sind.

Sobald OnValidate aufgerufen wurde, wird der OnBrowse nicht mehr aufgerufen.

OnChangeProp (prop, val, dictionary)

Wird bei Änderung einer Eigenschaft im Neu- und Eigenschaften-Dialog aufgerufen.

Parameter	Beschreibung
prop	Eigenschaft auf die Reagiert werden soll.
val	Dazugehöriger Wert zur Eigenschaft.
dictionary	Übergibt das Dictionary Objekt

Rückgabe: 1 := Das Dictionary enthält Eigenschaften, die in den Dialog zu übernehmen sind.

OnPopulateNew(kBefore, dictionary)

Wird beim Öffnen des Neu-Dialogs vor dem Erstellen eines neuen Dokuments aufgerufen

Parameter	Beschreibung
kBefore	Zeitpunkt wann die Aktion ausgeführt wird (hier unrelevant)
dictionary	Übergibt das Dictionary Objekt

Rückgabe: 1 := Das Dictionary enthält Eigenschaften, die in den Dialog zu übernehmen sind.

container.*

Die Skriptdatei „container“ enthält Ereignisse, die im Neu- und Eigenschaften-Dialogs eines Ordners aufgerufen werden. Sobald Scriptdatei im Script-Ordner gefunden wurde, erscheint neben dem Ordnernamen eine Schaltfläche. Beispielecodes sind unter **../script/example/container.example.vbs** für sie bereitgelegt.

OnBrowse (kBefore, dictionary)

Wird durch betätigen der Schaltfläche rechts neben dem Ordnernamen aufgerufen

Parameter	Beschreibung
kBefore	Zeitpunkt wann die Aktion ausgeführt werden soll(Hier unrelevant)
dictionary	Übergibt das Dictionary Objekt

Rückgabe: true oder 1 := Das Dictionary enthält Eigenschaften, die in den Dialog zu übernehmen sind.

OnValidate(kBefore, dictionary)

Wird beim klicken der 3 Punkte neben dem Ordnername aufgerufen. Des weiteren kann die Funktion beim Bestätigen des Neu-Dialogs aufgerufen werden. Dazu muss die Eigenschaft **container.new.validate** muss aktiv sein. Die Funktion wird beim Neu erstellen oder beim ändern von Eigenschaften aufgerufen.

Parameter	Beschreibung
kBefore	Zeitpunkt wann die Aktion ausgeführt werden soll(Hier unrelevant)
dictionary	Übergibt das Dictionary Objekt

Rückgabe: 1 := Das Dictionary enthält Eigenschaften, die in den Dialog zu übernehmen sind.

Sobald ein OnValidate aufgerufen wurde, wird der OnBrowse nicht mehr aufgerufen.

swxSpeedy.*

Die Skriptdatei „swxSpeedy“ enthält Ereignisse, die vom Solidworks AddIn aufgerufen werden. Beispielecodes sind unter **../script/example/swxSpeedy.vbs** für sie bereitgelegt.

OnFileSave(lEventTiming, dictionary)

Ereignis-Routine beim Speichern.

Parameter	Beschreibung
lEventTiming	Zeitpunkt wann die Aktion ausgeführt werden soll
dictionary	Übergibt das Dictionary Objekt

OnFileSaveAs(lEventTiming, dictionary)

Ereignis-Routine beim Speichern unter.

Eigenschaft	Beschreibung
lEventTiming	Zeitpunkt wann die Aktion ausgeführt werden soll
dictionary	Übergibt das Dictionary Objekt

OnFileCheckIn(lEventTiming, dictionary)

Ereignis-Routine beim Einchecken.

Eigenschaft	Beschreibung
lEventTiming	Zeitpunkt wann die Aktion ausgeführt werden soll
dictionary	Übergibt das Dictionary Objekt

OnFileRelease(lEventTiming, dictionary)

Ereignis-Routine beim Freigeben.

Eigenschaft	Beschreibung
lEventTiming	Zeitpunkt wann die Aktion ausgeführt werden soll
dictionary	Übergibt das Dictionary Objekt

OnReleaseRecreate_SaveAs(lEventTiming, dictionary)

Ereignis-Routine zum Speichern beim Freigeben.

Eigenschaft	Beschreibung
lEventTiming	Zeitpunkt wann die Aktion ausgeführt werden soll
dictionary	Übergibt das Dictionary Objekt

OnPendingApprovalRecreate_SaveAs(lEventTiming, dictionary)

Ereignis-Routine zum Speichern bei Antrag auf Prüfung

Eigenschaft	Beschreibung
lEventTiming	Zeitpunkt wann die Aktion ausgeführt werden soll
dictionary	Übergibt das Dictionary Objekt

OnApproveRecreate_SaveAs(lEventTiming, dictionary)

Ereignis-Routine zum Speichern bei Prüfung

Eigenschaft	Beschreibung
lEventTiming	Zeitpunkt wann die Aktion ausgeführt werden soll
dictionary	Übergibt das Dictionary Objekt

OnFileOpen(lEventTiming, dictionary)

Ereignis-Routine die beim öffnen einer Datei ausgeführt wird.

Eigenschaft	Beschreibung
lEventTiming	Zeitpunkt wann die Aktion ausgeführt werden soll
dictionary	Übergibt das Dictionary Objekt

aisSpeedy.*

Die Skriptdatei „aisSpeedy“ enthält Ereignisse, die vom Inventor AddIn aufgerufen werden.
Beispielecodes sind unter **../script/example/aisSpeedy.vbs** für sie bereitgelegt.

OnFileSave(lEventTiming, dictionary)

Ereignis-Routine beim Speichern.

Eigenschaft	Beschreibung
lEventTiming	Zeitpunkt wann die Aktion ausgeführt werden soll
dictionary	Übergibt das Dictionary Objekt

OnFileSaveAs(lEventTiming, dictionary)

Ereignis-Routine beim Speichern unter.

Eigenschaft	Beschreibung
lEventTiming	Zeitpunkt wann die Aktion ausgeführt werden soll
dictionary	Übergibt das Dictionary Objekt

OnFileCheckIn(lEventTiming, dictionary)

Ereignis-Routine beim Einchecken

Eigenschaft	Beschreibung
lEventTiming	Zeitpunkt wann die Aktion ausgeführt werden soll
dictionary	Übergibt das Dictionary Objekt

OnFileRelease(lEventTiming, dictionary)

Ereignis-Routine beim Freigeben

Eigenschaft	Beschreibung
lEventTiming	Zeitpunkt wann die Aktion ausgeführt werden soll
dictionary	Übergibt das Dictionary Objekt

OnReleaseRecreate_SaveAs(lEventTiming, dictionary)

Ereignis-Routine zum Speichern beim Freigeben

Eigenschaft	Beschreibung
lEventTiming	Zeitpunkt wann die Aktion ausgeführt werden soll
dictionary	Übergibt das Dictionary Objekt

OnPendingApprovalRecreate_SaveAs(lEventTiming, dictionary)

Ereignis-Routine zum Speichern bei Antrag auf Prüfung

Eigenschaft	Beschreibung
lEventTiming	Zeitpunkt wann die Aktion ausgeführt werden soll
dictionary	Übergibt das Dictionary Objekt

OnApproveRecreate_SaveAs(lEventTiming, dictionary)

Ereignis-Routine zum Speichern bei Prüfung

Eigenschaft	Beschreibung
lEventTiming	Zeitpunkt wann die Aktion ausgeführt werden soll
dictionary	Übergibt das Dictionary Objekt

OnSaveAs(lEventTiming, dictionary)

Ereignis-Routine zum Speichern in anderen Dateiformaten. Wird z.B. vom speedy-Zusatzbefehl „Speichern unter“ verwendet.

Eigenschaft	Beschreibung
lEventTiming	Zeitpunkt wann die Aktion ausgeführt werden soll
dictionary	Übergibt das Dictionary Objekt

OnFileOpen(lEventTiming, dictionary)

Ereignis-Routine die beim öffnen einer Datei ausgeführt wird.

Eigenschaft	Beschreibung
lEventTiming	Zeitpunkt wann die Aktion ausgeführt werden soll
dictionary	Übergibt das Dictionary Objekt

From:
https://wiki.speedy-pdm.de/ - **speedyPDM - Wiki**

Permanent link:
https://wiki.speedy-pdm.de/doku.php?id=speedy:30_modules:script

Last update: **2024/01/17 09:02**