

Tabellen lesen mit SAP® GUI Scripting

von Stefan Schnell

SAP® GUI Scripting bietet sehr viele Möglichkeiten und Ansätze. Neben der Simulation von Benutzereingaben können mit SAP® GUI Scripting noch ganz andere Ansätze realisiert und Perspektiven eröffnet werden. Allerdings ist es dazu notwendig die Möglichkeiten von Microsoft® Visual Basic® Script mit denen von SAP® GUI Scripting geschickt zu kombinieren. Im Folgenden wird beschrieben wie mit SAP® GUI Scripting der Transaktionscode SE16 ausgeführt und eine beliebige Tabelle geladen wird. Die Anzeige wird in den ALV-Grid-Modus (ABAP® List Viewer) geschaltet und es wird Zeile für Zeile ausgelesen. Die gelesenen Zeilen werden dann von Microsoft® Visual Basic® Script in eine Datei geschrieben. Die Felder werden mit einem Semikolon voneinander getrennt, so dass die so entstandene CSV-Datei am Ende z.B. von Microsoft® Excel eingelesen und weiterverarbeitet werden kann.

Um den Aufruf innerhalb des SAP® GUI Scripting einfach zu halten, definieren wir eine Sub-Routine mit dem Namen `ReadTableInFile`. Als Parameter der Sub-Routine übergeben wir den Tabellennamen und den Dateinamen.

```
'-ReadTableInFile-----
Sub ReadTableInFile(TableName, FileName)

End Sub
```

Vorgeplänkel

Die Sub-Routine werden wir nun Schritt für Schritt ausbauen. Wir beginnen mit dem Zurücksetzen der Session, in welcher das SAP® GUI Script ausgeführt wird. Dieser Schritt ist so eigentlich nicht direkt notwendig, aber für den Einstieg ist das Schaffen einer definierten Basis immer eine gute Voraussetzung. Dann wird der Transaktionscode SE16 ausgeführt und die gewünschte Tabelle zur Anzeige gebracht.

```
'-Zurücksetzen der Session-----
session.findById("wnd[0]/tbar[0]/okcd").text = "/n"
session.findById("wnd[0]/tbar[0]/btn[0]").press

'-Ausführung des Transaktionscodes SE16-----
session.findById("wnd[0]/tbar[0]/okcd").text = _
"/nSE16"
session.findById("wnd[0]/tbar[0]/btn[0]").press

'-Tabelle anzeigen-----
session.findById("wnd[0]/usr/ctxtDATABROWSE-TABLENAME").text = _
TableName
session.findById("wnd[0]/tbar[1]/btn[7]").press
session.findById("wnd[0]/tbar[1]/btn[8]").press
```

Aktivierung des ALV-Grid

Nun aktivieren wir die ALV-Grid-Anzeige der Tabelle. Da diese Einstellungen über das Menü des SAP® GUI vorgenommen werden, sind diese sprachabhängig. Der hier vorgestellte Ansatz bezieht sich auf einen SAP® GUI in deutscher Sprache.

```
'-Aktiviert die Anzeige als ALV-Grid-----
Set Menu = session.findById("wnd[0]/mbar")
Set Einstellungen = Menu.FindByName("Einstellungen", _
"GuiMenu")
Set BenutzerPar = _
Einstellungen.FindByName("Benutzerparameter...", _
"GuiMenu")
BenutzerPar.Select()

Set ALVGridView = session.findById("wnd[1]/usr/" & _
"tabsG_TABSTRIP/tabp0400/ssubTOOLAREA:" & _
"SAPLWB_CUSTOMIZING:0400/radrSEUMOD-TBALV_GRID")
```

```
If ALVGridView.Selected = vbFalse Then
    ALVGridView.select()
End If
session.findById("wnd[1]/tbar[0]/btn[0]").press

Set BenutzerPar = Nothing
Set Einstellungen = Nothing
Set Menu = Nothing
```

Wird der SAP GUI in einer anderen Sprache als Deutsch verwendet, so sind die Zeilen mit `FindByName` entsprechend anzupassen.

Nachgeplänkel

Bevor es an das Schreiben geht, ermitteln wir zuerst die Anzahl der Spalten und Zeilen der Tabelle. Die Ermittlung dieser Zahlen ist für das Lesen der Tabelle bzw. das Schreiben in die CSV-Datei von besonderer Bedeutung, da sie den Rahmen der Lese- und Schreibschleife definieren.

```
Set table = session.findById( _
"wnd[0]/usr/ctrlGRID1/shellcont/shell")
Rows = table.RowCount() - 1
Cols = table.ColumnCount() - 1
```

Jetzt wird geschrieben

Der nun folgende Abschnitt schreibt die Daten der angezeigten Tabelle in eine Datei. Dazu wird zuerst eine Textdatei erzeugt. Dann werden die Spalten-titel ermittelt und in diese Datei geschrieben. Nun werden die Tabellendaten Zeile für Zeile und Spalte für Spalte gelesen und ebenfalls in die Datei geschrieben. Für die Trennung der einzelnen Felder wird ein Semikolon verwendet. Nach jeder Zeile erfolgt ein Zeilenumbruch und nach 32 Zeilen wird das ALV-Grid aktualisiert. Zu guter Letzt wird die Textdatei noch geschlossen.

```
'-Schreibe Tabelle in eine CSV-Datei-----
Set oFile = CreateObject("Scripting.FileSystemObject")
If IsObject(oFile) Then
    Set CSVFile = oFile.CreateTextFile(FileName, True)
    If IsObject(CSVFile) Then

'-Ermittlung der Spaltentitel-----
        Set Columns = table.ColumnOrder()
        For j = 0 To Cols
            If j = Cols Then
                CSVFile.Write(CStr(Columns(j)))
            Else
                CSVFile.Write(CStr(Columns(j)) & ";")
            End If
        Next
        CSVFile.WriteLine("")
```

Tabellen lesen mit SAP® GUI Scripting

```

For i = 0 To Rows
  For j = 0 To Cols
    If j = Cols Then
      CSVFile.Write(table.GetCellValue(i, _
        CStr(Columns(j))))
    Else
      CSVFile.Write(table.GetCellValue(i, _
        CStr(Columns(j))) & ";")
    End If
  Next
  '-Nach 32 Zeilen Grid aktualisieren-----
  If i Mod 32 = 0 Then
    table.SetCurrentCell i, CStr(Columns(0))
  End If
  '-Carriage und Return nach einer Zeile-----
  If i <> Rows Then
    CSVFile.WriteLine("")
  End If
Next
  CSVFile.Close
End If
End If

Set ALVGridView = Nothing
Set Columns = Nothing
Set table = Nothing

```

Der Aufruf

Zum Schluss erfolgt jetzt noch der Aufruf der Sub-Routine. Hier als Beispiel das Lesen der Tabelle SFLIGHT in die Datei SFlight.csv im Verzeichnis C:\Dummy.

```

'-Read the table SFLIGHT in a file-----
ReadTableInFile "SFLIGHT", "C:\Dummy\SFlight.csv"

```

Fazit

Wie gezeigt, können durch eine geschickte Kombination von SAP® GUI Scripting und Microsoft® Visual Basic® Script sehr interessante und praxisnahe Ergebnisse erzielt werden. Mit diesem Ansatz ist es nun sehr einfach möglich eine Reihe von Tabellen durch den Aufruf von ReadTableInFile zu erhalten. Dies kann sehr hilfreich sein, wenn z.B. Tabellen unterschiedlicher SAP®-Systeme verglichen oder Tabellen für weitere lokale Analysen z.B. in Microsoft® Access® verwendet werden sollen.

Haben Sie Fragen, Anregungen oder Anmerkungen, so senden Sie einfach eine E-Mail an mail@stschnell.de.

Warenzeichen

- SAP, ABAP und SAP GUI Scripting sind eingetragene Warenzeichen der SAP AG
- Microsoft, Visual Basic Scripting, Access und Excel sind eingetragene Warenzeichen der Microsoft Corporation, USA

Links

- www.stschnell.de

Tabellen lesen mit SAP® GUI Scripting

```

'-Begin-----
'-ReadTableInFile-----
  Sub ReadTableInFile (TableName, FileName)

'-Reset the session-----
  session.findById("wnd[0]/tbar[0]/okcd").text = "/n"
  session.findById("wnd[0]/tbar[0]/btn[0]").press

'-Open TAC SE16-----
  session.findById("wnd[0]/tbar[0]/okcd").text = "/nSE16"
  session.findById("wnd[0]/tbar[0]/btn[0]").press

'-View table-----
  session.findById("wnd[0]/usr/ctxtDATABROWSE-TABLENAME").text = _
    TableName
  session.findById("wnd[0]/tbar[1]/btn[7]").press
  session.findById("wnd[0]/tbar[1]/btn[8]").press

'-Set display to ALV Grid view-----

'-Open user specific parameters dialog-----
'_
'- Attention: Here is a language specific code
'_
-----
  Set Menu = session.findById("wnd[0]/mbar")
  Set Einstellungen = Menu.FindByName("Einstellungen", "GuiMenu")
  Set BenutzerPar = Einstellungen.FindByName("Benutzerparameter...", _
    "GuiMenu")
  BenutzerPar.Select()

'-Set the display-----
  Set ALVGridView = session.findById("wnd[1]/usr/tabsG_TABSTRIP/" & _
    "tabp0400/ssubTOOLAREA:SAPLWB_CUSTOMIZING:0400/radrSEUMOD-TBALV_GRID")
  If ALVGridView.Selected = vbFalse Then
    ALVGridView.select()
  End If
  session.findById("wnd[1]/tbar[0]/btn[0]").press

  Set BenutzerPar = Nothing
  Set Einstellungen = Nothing
  Set Menu = Nothing

'-Get rows and columns-----
  Set table = session.findById("wnd[0]/usr/ctrlGRID1/shellcont/shell")
  Rows = table.RowCount() - 1
  Cols = table.ColumnCount() - 1

```

```

'-Write the table to a CSV file-----
Set oFile = CreateObject("Scripting.FileSystemObject")
If IsObject(oFile) Then
    Set CSVFile = oFile.CreateTextFile(FileName, True)
    If IsObject(CSVFile) Then

        '-Get the title of all columns in the first line-----
        Set Columns = table.ColumnOrder()
        For j = 0 To Cols
            If j = Cols Then
                CSVFile.Write(CStr(Columns(j)))
            Else
                CSVFile.Write(CStr(Columns(j)) & ";")
            End If
        Next
        CSVFile.WriteLine("")

    For i = 0 To Rows
        For j = 0 To Cols
            If j = Cols Then
                CSVFile.Write(table.GetCellValue(i, _
                    CStr(Columns(j))))
            Else
                CSVFile.Write(table.GetCellValue(i, _
                    CStr(Columns(j))) & ";")
            End If
        Next

        '-Each 32 lines actualize the grid-----
        If i Mod 32 = 0 Then
            table.SetCurrentCell i, CStr(Columns(0))
        End If

        '-Carriage and return after a line-----
        If i <> Rows Then
            CSVFile.WriteLine("")
        End If

    Next

    CSVFile.Close
End If
End If

Set ALVGridView = Nothing
Set Columns = Nothing
Set table = Nothing

End Sub

```

Tabellen lesen mit SAP® GUI Scripting

```
'-Main-----  
  If Not IsObject(application) Then  
    Set SapGuiAuto = GetObject("SAPGUI")  
    Set application = SapGuiAuto.GetScriptingEngine  
  End If  
  
  If Not IsObject(connection) Then  
    Set connection = application.Children(0)  
  End If  
  
  If Not IsObject(session) Then  
    Set session = connection.Children(0)  
  End If  
  
  '-Read the table SFLIGHT in a file-----  
  ReadTableInFile "SFLIGHT", "C:\\Dummy\\SFlight.csv"  
  
'-End-----
```