

### 17.7.6 GridView – Daten-Export in eine csv-Datei

In diesem Projekt geht es in erster Linie nicht um das Anzeigen von Daten in einer GridView, sondern um die Sicherung von Daten, die in einer GridView angezeigt werden. Gesichert werden die Daten in einer csv-Datei.

Dieses Textdatei-Format gilt als Austauschformat für einfach strukturierte Daten. Jede Zeile nimmt einen Datensatz auf, wobei die einzelnen Daten-Felder mit einem Feldtrennzeichen von den anderen Feldern getrennt werden und ein Feldwert von einem Texttrennzeichen eingeschlossen wird. Optional kann die erste Zeile auch die Feldbezeichner aufnehmen.

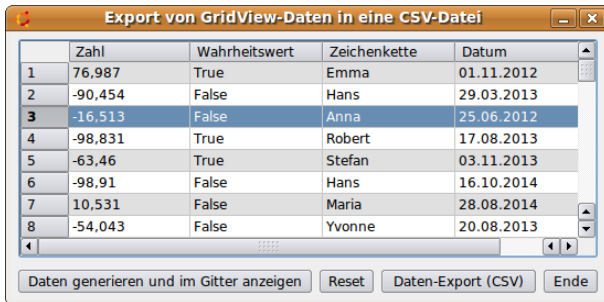


Abbildung 17.7.6.1: Speicherung von Daten in einer csv\_Datei

Hier sehen Sie einen Ausschnitt mit den in einer csv-Datei gespeicherten Daten – vergleichen Sie bitte mit Abbildung 17.7.6.1 – bei der im ersten Datensatz die 4 Feldnamen gespeichert sind:

```
"Zahl", "Wahrheitswert", "Zeichenkette", "Datum"  
"76,987", "True", "Emma", "01.11.2012"  
"-90,454", "False", "Hans", "29.03.2013"  
"-16,513", "False", "Anna", "25.06.2012"  
"-98,831", "True", "Robert", "17.08.2013"  
"-63,46", "True", "Stefan", "03.11.2013"
```

Als Feldtrennzeichen dient das Komma und als Texttrennzeichen wird das Anführungszeichen " eingesetzt. Sie können im Export-Formular sowohl das Feldtrennzeichen als auch das Texttrennzeichen aus einer vorgegebenen Zeichen-Menge auswählen. Ob im ersten Datensatz die Feldnamen gespeichert werden sollen, können Sie auch festlegen.



Abbildung 17.7.6.2: Festlegung von Speicheroptionen in einem eigenen Formular

Den Speicherort und den Dateinamen für die csv-Datei bestimmen Sie in einem Dialog:



Abbildung 17.7.6.3: Speichern-Dialog

Um den Quelltext für das Speichern von Daten in einer csv-Datei auch in anderen Projekten zu verwenden, wurden ein eigenes Formular zum Festlegen der Speicheroptionen und ein Modul verwendet.

Der vollständigen Quelltext für das Modul finden Sie hier:

```
' Gambas module file

PUBLIC sFeldTrennzeichen AS String
PUBLIC sTextTrennzeichen AS String
PUBLIC bSaveTitel AS Boolean
PUBLIC bError AS Boolean

PUBLIC SUB ExportGridView2CSV(Grid AS GridView,pFeldTrenner AS String, pTextTrenner AS String, /
    pSaveTitel AS Boolean)

    DIM iCount, rCount, cCount AS Integer
    DIM sZeile AS String
    DIM hFile AS File

    Dialog.Title = "Speichern Sie die Daten in einer csv-Datei!"
    Dialog.Filter = [ "*.csv", "csv-Dateien" ]
    IF Dialog.SaveFile() = TRUE THEN
        Message.Warning("Der Daten-Export wurde abgebrochen!")
        RETURN ' Cancel-Button
    ELSE
        ' Die CSV-Datei wird neu angelegt oder geleert
        IF File.Ext(Dialog.Path) <> "csv" AND File.Ext(Dialog.Path) <> "txt" THEN
            Dialog.Path = File.SetExt(Dialog.Path, "csv")
        ENDIF
        hFile = OPEN Dialog.Path FOR WRITE CREATE
        IF ERROR THEN
            Message.Error(Error.Text & Error.Where)
            RETURN
        ENDIF ' ERROR
        SEEK #hFile, 0

        ' Spaltenüberschriften speichern (optional)
        IF pSaveTitel = TRUE THEN
            IF Grid.Header = Grid.Horizontal OR Grid.Header = Grid.Both THEN
                FOR iCount = 0 TO Grid.Columns.Count - 1
                    IF iCount > 0 THEN
                        sZeile &= pFeldTrenner & pTextTrenner & Grid.Columns[iCount].Title & pTextTrenner
                    ELSE
                        sZeile &= pTextTrenner & Grid.Columns[iCount].Title & pTextTrenner
                    ENDIF ' iCount > 0
                NEXT
                PRINT #hFile, sZeile
            ENDIF ' Grid.Horizontal OR Grid.Header?
        ENDIF ' bTitel = TRUE?

        ' GridView-Daten speichern
        sZeile = ""
        FOR rCount = 0 TO Grid.Rows.Count - 1
            FOR cCount = 0 TO Grid.Columns.Count - 1
                IF cCount > 0 THEN
                    sZeile &= pFeldTrenner & pTextTrenner & Grid[rCount, cCount].Text & pTextTrenner
                ELSE
                    sZeile &= pTextTrenner & Grid[rCount, cCount].Text & pTextTrenner
                ENDIF ' cCount > 0
            NEXT ' cCount
            PRINT #hFile, sZeile
            sZeile = ""
        NEXT ' rCount
    ENDIF ' Dialog.SaveFile() = TRUE?

    CLOSE #hFile

END ' GridViewExport2CSV
```

Im Hauptprogramm ist vor allem die Prozedur interessant, mit der ein Daten-Export angeschoben wird:

```
PUBLIC SUB ExportGridView2CSV()
    FExport.ShowModal
    IF MCSV.bError = FALSE THEN
        MCSV.GridView2CSV(GridView1, MCSV.sFeldTrennzeichen, MCSV.sTextTrennzeichen, MCSV.bSaveTitel)
        btnGridViewExportCSV.Enabled = TRUE
    ELSE
        Message.Warning("Der Daten-Export wurde abgebrochen!")
        btnGridViewExportCSV.Enabled = TRUE
    ENDIF ' MCSV.bError = FALSE?
END ' ExportGridView2CSV()
```