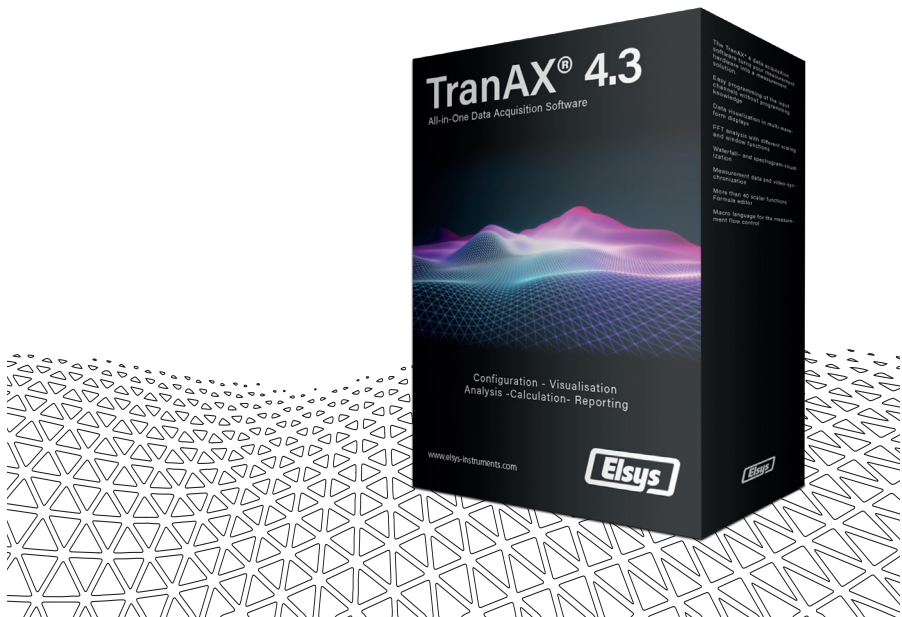


# TranAX 4


Report-Generator

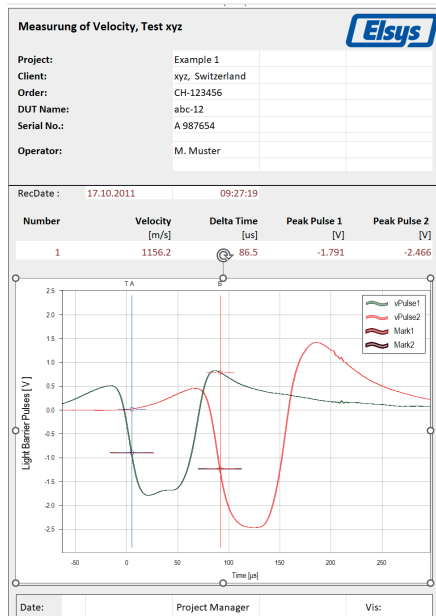


Benutzerhandbuch

# 1. TranAX Report Generator

Der Report Generator basiert auf speziellen Funktionen im Formel Editor und ermöglicht einen Messreport als Excel Datei zu erstellen.

 Diese Funktionen stehen nur bei aktivierter Report-Generator Option zu Verfügung.



## 1.1 Einleitung

Die Basis eines Reportes ist eine bereits existierende Excel Datei welches als Template verwendet wird. Formatierungen wie Schriftgrößen, Zellfarben oder Zellränder werden in diesem Template in Excel definiert. Mit Hilfe des Formel Editores kann dieses Template mit Messkurven oder Skalarwerten abgefüllt werden.

Der Zugriff auf die Excel Datei kann auf zwei verschiedene Arten geschehen: Befindet sich Excel auf dem System, kann über die COM Schnittstelle auf die Excel Datei zugegriffen werden. Dies bietet den Vorteil, dass der Report gleich geöffnet wird und die eingefügten Daten sofort sichtbar sind. Die zweite Variante

greift direkt auf die Datei zu. Hierzu wird kein installiertes Excel benötigt.

Als Template Dateien können entweder Excel Dateien im Format .xlsx oder .xltx verwendet werden. Es wird empfohlen die xltx Dateien als Vorlagen zu verwenden, da diese auch nicht so schnell irrtümlicher Weise verändert werden können.

## 1.2 Report Funktionen

### 1.2.1 Initialisierung und Dateifunktionen

XlInitializeCOM()	Einer der beiden Befehle muss ausgeführt werden bevor andere Report-Generator-Funktionen aufgerufen werden können.
XlInitializeXML()	<p>XlInitializeCOM() baut eine Verbindung zu Excel über die COM Schnittstelle auf. (Excel muss installiert sein auf dem Gerät)</p> <p>XlInitializeXML() baut eine Verbindung direkt über die Datei auf. Es wird kein Excel benötigt.</p>
XlCreateEmptyFile()	<p>Erstellt eine leere Excel Datei und gibt einen zufälligen Namen mit der Endung «.xlsx» als Rückgabewert zurück. Dieser Name muss weiterverwendet werden um auf der Datei arbeiten zu können. Per default ist dieser Datei der Pfad «data» des aktuellen Experiments zugewiesen.</p> <p>Diese Funktion empfiehlt sich, wenn errechnete Werte schnell in eine Excel-Datei gespeichert werden sollen. Sie beruht nicht auf einer zuvor erstellen Vorlage, kann aber im Nachhinein mit Excel weiterverarbeitet werden.</p> <p>Wenn diese Funktion aufgerufen wird, kann XlOpenFile () entfallen.</p> <p>Beispiel:</p> <pre>fn = XlCreateEmptyFile() fn2 = "MyRreport_#.xlsx" ; any own name XlSetCellValue(fn, "A10", 7.123) XlSave(fn, fn2) ; fn2 will be saved in «.\data»</pre>
XlOpenFile (fname)	<p>Öffnet eine bestehende Excel-Datei mit dem Namen fname. Als Rückgabewert erhält man wieder den Datei-Namen ergänzt mit dem absoluten Datei-Pfad. Dieser muss in nachfolgenden Excel-Funktionen verwendet werden.</p> <p>Beispiel:</p> <pre>XlInitializeXML() fn = ".\Excel_Templates\MyTemplate.xlsx" filename = XlOpenFile(fn) XlSetCellValue(filename, "A10", 7)</pre>

**XISave (fname, fnameReport)** Speichert die offene Excel Datei «fname» inkl. allen Änderungen.

Wenn die Vorlage nicht überschrieben werden soll, muss für den zweiten Parameter «fnameReprot» ein anderer Namen (inkl. Pfad) gewählt werden.

Solange danach kein CloseFile(fname)-Befehl ausgeführt wird, bleibt die Excel-Mappe geöffnet. Es könnten also unmittelbar zusätzliche Einträge gemacht werden.

Beispiel:

```
XIInitializeCOM()
fn = ".\Excel_Templates\MyTemplate.xlsx"
fnam = XLOpenFile(fn)
fnRep = ".\Excel_Reports\Report_#.xlsx"
; Block with value insertions in EXCEL Sheet.
XISave (fn, fnRep) ; Saves the EXCEL file (incl. all insertions) in the Report File.
```

## 1.2.2 Excel-Blatt Funktionen

**XICreateSheet(fname [,name [templateName]])** Erzeugt ein neues Blatt innerhalb einer Excel-Datei. Der Parameter «fname» vom Typ String ist der aktuell verwendete Name der Datei.

Der erste optionale Parameter «name» vom Typ String definiert den Namen des neuen Blattes. Ohne diesen Parameter erhält das neue Blatt den Namen «Empty» plus eine aufsteigende Zahl, z.B. «Empty 5»

Der zweite optionale Parameter «templateName» vom Typ String kann verwendet werden, um ein bestehendes Blatt innerhalb der Excel-Datei zu kopieren und mit einem neuen Namen zu erstellen.

**XISelectSheet (fname, sheet)** Wählt in der Excel Datei «fname» das Tabellen-Blatt mit dem Name «sheet». Wenn dieser Befehl fehlt, wird standardmässig das erste Blatt gewählt.

Beispiel:

```
fn = ".\Excel_Templates\MyTemplate.xlsx"
filename = XLOpenFile(fn)
XISelectSheet(filename, "Tabelle2")
```

XLRemoveSheet(fname,  
sheetname)

Entfernt das Blatt «sheetname» aus der Datei.

XLRenameSheet(fname,  
sheetname, newSheetname)

Benennt ein Excel Blatt um.

### 1.2.3 Zellfunktionen

XLGetCellValue (fname, cell)

Gibt den Wert in der Zelle «cell» zurück.

In der Regel ist «cell» eine einzelne Zelle («B3»).

Als «cell» wird ein String erwartet. Je nach dem was in der Zelle eingetragen ist, wird ein String oder eine Zahl zurück gegeben.

Es kann auch ein Zellenbereich definiert werden (z.B. «A1:A4» oder «A1:D1»). Der Zellenbereich darf jeweils nur in derselben Kolonne oder Zeile definiert sein (nicht z.B. «A1:C3»). Auch muss vorher ein entsprechendes Array oder eine Liste deklariert worden sein. Es wird dann das Array oder die Liste abgefüllt mit den String- oder Zahlen-Werten der Zellen. Arrays müssen als String-Array oder Double-Array deklariert werden, je nachdem ob Strings oder Zahlen in den Zellen erwartet werden.

Wenn eine Zelle leer ist, wird ein Leerstring bzw. «n. def.» zurück gegeben. Wenn eine Liste deklariert wurde, enthalten danach deren Elemente Strings oder Zahlen

Beispiel:

```
fn = ".\Excel_Templates\Template_1.xlsx"
```

```
fnam = XLOpenFile(fn)
```

```
v1 = XLGetCellValue(fnam, "A1") ; single value
```

```
valArr = Array(0 to 3) as String
```

```
valArr = XLGetCellValue(fnam, "A2:A5")
```

```
valArrD = Array(0 to 3) as Double
```

```
valArrD = XLGetCellValue(fnam, "B2:E2")
```

```
valArrL = List()
```

```
valArrL = XLGetCellValue(fnam, "A2:E2")
```

XIGetCellReferences  
(fname, searchKey [, all] )

Die Funktion sucht in der ganzen Excel-Datei «fname» nach dem Schlüssel «searchKey» und gibt die gefundene Zellen-Bezeichnung zurück (z.B. «B5»). Diese Parameter müssen vom Typ String sein.

Schlüsselwörter müssen entsprechend vorher in der Excel-Mappe (Vorlage) eingetragen worden sein. Falls «keyword» nicht gefunden wird, bringt die Funktion «n. def.» zurück. Es empfiehlt sich, Schlüsselwörter zu wählen, die sonst kaum vorkommen (z.B. «%V1»). Zwischen Klein- und Grossbuchstaben wird kein Unterschied gemacht.

Falls der optionale Parameter all = True angegeben wird, wird ein Array bzw. eine Liste zurück gegeben. Darin sind alle Zellen-Bezeichnungen eingetragen, bei welchen das Schlüsselwort searchKey gefunden wurde.

Beispiel:

```
fn = ".\Excel_Templates\MyTemplate.xlsx"
fnam = XIOpenFile(fn)
cellRef = XIGetCellReferences(fnam, "%x")
```

XIChangeColumn (cell,  
number)

Addiert die angegebene Zahl «number» zu der definierten Zelle «cell». Dabei wird die Kolonne von «cell» beeinflusst. «cell» muss ein String (z.B. «B5») und «number» eine Zahl sein.

Beispiel:

```
cell = "B2"
newCell = XIChangeColumn(cell, 2) ; Ergibt «D2»
```

XIChangeRow (cell, number)

Addiert die angegebene Zahl «number» zu der definierten Zelle «cell». Dabei wird die Zeile (row) von «cell» beeinflusst. «cell» muss ein String (z.B. «B5») und «number» eine Zahl sein.

Beispiel:

```
cell = "B2"
newCell = XIChangeRow(cell, 2) ; Ergibt «B4»
```

XISetCellValue  
(fname, cell | searchKey,  
value)

Setzt in der Excel-Datei «fname» den Wert «value» in die Zelle «cell». Anstelle einer Zelle (z.B. «B2») kann auch ein Schlüssel «searchKey» vorgegeben werden (dadurch kann XIGetCellReferences entfallen).

«fname» und «cell» bzw. «searchKey» müssen Strings sein. «value» kann eine Zahl, String, Array, Liste oder Kurve sein. Wenn «value» ein Array, Liste oder eine Kurve ist, werden die Werte standardmässig in die Kolonne der spezifizierten Zelle «cell» eingefügt.

Möchte man die Werte in eine Zeile einfügen, muss die Zelle folgendermassen definiert werden: «A7:row».

Kurven (wie auch Arrays oder Listen) dürfen nicht zu lang sein für die Übertragung in Excel.

Die maximale Länge (Anzahl Samplings, bzw. Elemente) beträgt:

Übertragung in eine Kolonne: 1'048'577 - Linie des ersten Wertes (z.B. <=1'048'571, wenn cell = "x6").

Übertragung in eine Zeile: 16'385 - Kolonne des ersten Wertes (z.B. <=16'382, wenn cell = "Cx:row").

Zu lange Kurven (grosse Blocklängen) können vorher mit Slice(..), Resampling(..) oder Skip(..) verkürzt werden.

Beispiel:

```
filename = "excel\Template.xlsx"
```

```
filename = XIOpenFile(filename)
```

```
colArr = Array(1,2,3,4)
```

```
XISetCellValue(filename, "A1", colArr)
```

```
XISetCellValue(filename, "A1:row", colArr)
```

XIEnd (fname, cell, direction)  
(ab TranAX 4.3 und neuer)

Sucht die letzte nicht leere Zelle ab der Zelle «cell» und in Richtung «direction».

Richtungen:

- 0 = nach oben
- 1 = nach unten
- 2 = nach links
- 3 = nach rechts

## 1.2.4 Bildfunktionen

`XlInsertImage (fname, winname,  
cell [, width [, height ]])`

Fügt einen Screenshot des Fensters «winname» an der angegebenen Position cell in der Excel-Mappe «fname» ein. «winname» muss ein String sein. Dies kann der Name einer einzelnen Kurvenanzeige, einer Skalar-Tabelle, einer ganzen Seite oder einer Bild-Datei (inkl. Pfadangaben) sein. «cell» ist in der Regel eine einzelne Zelle (z.B. «B2»). Sie bestimmt die linke obere Ecke des einzufügenden Bildes. Für «cell» kann auch ein Bereich angegeben werden (z.B. «A4:F30»). Der Zellen-Bereich bestimmt dabei den Ort und die Grösse des eingefügten Bildes. Fakultativ kann die Breite «width» und Höhe «height» des eingefügten Bildes definiert werden. Wird nur die Breite angegeben, so wird die Höhe proportional dazu bestimmt (gleiche Seitenverhältnisse wie beim Original-Bild). «width» und «height» sind in Anzahl Pixels anzugeben. Wenn ein Zellen-Bereich (nicht nur eine Einzel-Zelle) vorgegeben ist, werden «width» und «height» ignoriert.

Wenn unter «Extras / Einstellungen / Benutzeroberfläche / Snapshot» Weisser Hintergrund» markiert ist, besitzt das übertragene Bild immer eine entsprechende Hintergrund-Farbe. Falls dabei auf dem Bildschirm mit schwarzem Hintergrund gearbeitet wird, sollten die Kurven-Farben eher dunkel gewählt werden.

Beispiel:

```
fn = ".\Excel_Templates\MyTemplate.xlsx"  
fnam = XlOpenFile(fn)  
XlInsertImage(fnam, "Waveform 1", "A4:E20")  
XlInsertImage(fnam, "Skalar_A 1", "A21", 100, 300)
```

## 1.2.5 Kopf und Fusszeilen Funktionen

XISetHeader (fname, alignment | searchKey, value)

Setzt in der Kopf- bzw. Fuss-Zeile der EXCEL-Mappe «fname» den Wert «value» ein.

XISetFooter (fname, alignment | searchKey, value)

Der zweite Parameter bestimmt, wo «value» eingefügt werden soll. Dieser kann ein Schlüsselwort XILeft, XICenter, XIRight (für Links, Zentriert, Rechts) oder ein Suchwort sein. Ein Suchwort muss vorher in der Kopf- bzw. Fuss-Zeile der Excel-Vorlage eingefügt worden sein.

Beispiel:

```
Fn = "excel\MyTemplate.xlsx"
```

```
Fn = XIOpenFile(Fn)
```

```
Skey="%Hd-1" ; Should be set in Template
```

```
XISetHeader(Fn, XILeft, "Report")
```

```
XISetHeader(Fn, Skey, "Unit Test 1")
```

```
XISetFooter(Fn, XICenter, "Result Table")
```

XILeft

XICenter

XIRight

Schlüsselwörter für den Parameter alignment der Funktionen XISetHeader() bzw. XISetFooter()