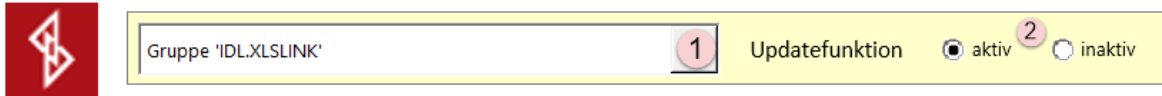


IDL.XLSLink – macro capability

Calling functions via VBA-Macro

Spreadsheet 'ID functions'

A routine is stored here that is can execute the various functions.



Click this button to start the procedure.



1. Choose the appropriate function of IDL.XLSLINK.
 - The function will not start unless certain preconditions are met.
 - Please see the according Excel 'DisplayAlerts'.
2. Options for the update-function.
 - Updatefunktion active = XLSLINK Update Function (Open Update, 201) will be executed.
 - Updatefunktion inactive = XLSLINK Update Function (Block Update, 200) will be executed.
 - If for example, a workbook is to be updated, the update function must be active, otherwise the IDL formulas will not be calculated and displayed correctly.

Preconditions for certain functions.:

CallID	XLSLink - Funktion	Beschreibung	Bezeichnung XLSLink
	Gruppe 'Aktualisieren'		
2	selektierter Bereich	Bereich aktualisieren	BereichAktualisieren
3	Tabellenblatt	Arbeitsblatt aktualisieren	ArbeitsblattAktualisieren
4	Arbeitsmappe	Gesamtes Dokument aktualisieren	DokumentAktualisieren
	Feld/Auswahlliste v		
38	selektierter Bereich	IDLCombo-Bereich aktualisieren	IDLCombo_BereichAktualisieren
39	Tabellenblatt	IDLCombo-Arbeitsblatt aktualisieren	IDLCombo_ArbeitsblattAktualisieren
40	Arbeitsmappe	IDLCombo-Gesamtes Dokument aktualisieren	IDLCombo_DokumentAktualisieren
27	Protokoll	Fortschritt zeigen (wieviele offenen bzw beantwortete Anfragen)	ShowProgressInfo
	Gruppe 'Archiv'		
	Verpacken v		
19	selektierter Bereich	Alle markierten Formeln verpacken	PackAllFormulasCurrentSelection
20	Tabellenblatt	Alle Formeln im aktuellen Arbeitsblatt verpacken	PackAllFormulasCurrentWorksheet
21	Arbeitsmappe	Alle Formeln in Arbeitsmappe verpacken	PackAllFormulasCurrentWorkbook
	Entpacken v		
22	selektierter Bereich	Alle markierten Formeln entpacken	UnpackAllFormulasCurrentSelection
23	Tabellenblatt	Alle Formeln im aktuellen Arbeitsblatt entpacken	UnpackAllFormulasCurrentWorksheet
24	Arbeitsmappe	Alle Formeln in Arbeitsmappe entpacken	UnpackAllFormulasCurrentWorkbook
	Gruppe 'Export'		
7	selektierter Bereich	Bereich nach Konsis exportieren	ExportToKonsisCurrentSelection
8	Tabellenblatt	Tabellenblatt nach Konsis exportieren	ExportToKonsisCurrentWorksheet
9	Arbeitsmappe	Gesamte Arbeitsmappe nach Konsis exportieren	ExportToKonsisCurrentWorkbook
48	Export zurücksetzen	Alle IDLSetValue-Formeln zurücksetzen	ExportToKonsisResetFormula

- For these functions (highlighted in green), a maximum of one file with corresponding IDL formulas must/may be open.
- In order to be able to call the function "Update range" or "Update spreadsheet" by macro, for example, a corresponding range or spreadsheet must be selected in the second file.

List of all XLSLink - functions and IDs

CallID	XLSLink - Funktion	Beschreibung	Bezeichnung XLSLink
Gruppe 'IDL XLSLINK'			
1	Formeleditor aufrufen	Formel in AppConfig anzeigen	Edit_Bezug_Formel
5	Abfragen rücksetzen	Alle abfragen löschen	CancelAllRequests
17	Info-Anzeige	Excel Info-Anzeige aufrufen	ShowExcelLog
Gruppe 'Aktualisieren'			
2	selektierter Bereich	Bereich aktualisieren	BereichAktualisieren
3	Tabellenblatt	Arbeitsblatt aktualisieren	ArbeitsblattAktualisieren
4	Arbeitsmappe	Gesamtes Dokument aktualisieren	DokumentAktualisieren
Feld/Auswahlhilfe v			
38	selektierter Bereich	IDLCombo-Bereich aktualisieren	IDLCombo_BereichAktualisieren
39	Tabellenblatt	IDLCombo-Arbeitsblatt aktualisieren	IDLCombo_ArbeitsblattAktualisieren
40	Arbeitsmappe	IDLCombo-Gesamtes Dokument aktualisieren	IDLCombo_DokumentAktualisieren
27	Protokoll	Fortschritt zeigen (wieviele offenen bzw beantwortete Anfragen)	ShowProgressInfo
Gruppe 'Archiv'			
Verpacken v			
19	selektierter Bereich	Alle markierten Formeln verpacken	PackAllFormulasCurrentSelection
20	Tabellenblatt	Alle Formeln im aktuellen Arbeitsblatt verpacken	PackAllFormulasCurrentWorksheet
21	Arbeitsmappe	Alle Formeln in Arbeitsmappe verpacken	PackAllFormulasCurrentWorkbook
Entpacken v			
22	selektierter Bereich	Alle markierten Formeln entpacken	UnpackAllFormulasCurrentSelection
23	Tabellenblatt	Alle Formeln im aktuellen Arbeitsblatt entpacken	UnpackAllFormulasCurrentWorksheet
24	Arbeitsmappe	Alle Formeln in Arbeitsmappe entpacken	UnpackAllFormulasCurrentWorkbook
Gruppe 'Export'			
7	selektierter Bereich	Bereich nach Konsis exportieren	ExportToKonsisCurrentSelection
8	Tabellenblatt	Tabellenblatt nach Konsis exportieren	ExportToKonsisCurrentWorksheet
9	Arbeitsmappe	Gesamte Arbeitsmappe nach Konsis exportieren	ExportToKonsisCurrentWorkbook
48	Export zurücksetzen	Alle IDLSetValue-Formeln zurücksetzen	ExportToKonsisResetFormula
Datei v			
6	Kopie ohne Formeln erstellen	Tabelle in Text-Datei mit Zeichencode 1252 exportieren	ExportToCleanFile
Gruppe 'Werkzeuge'			
33	Nullzeichen ergänzen	Nullzeichen ergänzen	PadLeftCurrentSelection
47	Excel Passwort	Excel-Passwörter verwalten	EditExcelPassword
Feld/Auswahlhilfe v			
28	Einfügen aktive Zelle	Auswahlhilfe (IDLCombo) in Zelle einfügen	AddIDLCombo
29	entfernen	Auswahlhilfe (IDLCombo) aus Zelle löschen	RemoveIDLCombo
CallID	XLSLink - Funktion	Beschreibung	Bezeichnung XLSLink
Einstellungen (Zellinhalt) v			
Excel Berechnungsoption v			
25	Automatisch	Wechsel ExcelApplicationCalculationMode to xlCalculationAutomatic	ExcelCalculationModeAutomatic
26	Manuell	Wechsel ExcelApplicationCalculationMode to xlCalculationManual	ExcelCalculationModeManual
51	Semiautomatisch	Wechsel ExcelApplicationCalculationMode to xlCalculationSemiautomatic	ExcelCalculationModeSemiautomatic
35	Periode im Datumformat an/aus	Einstellung "Periode als Datumformat" ändern	TogglePeriodAsDateFormat
46	Lange Fehlermeldungen	Fehlertexte in Excel Lang oder Kurz zeigen	ToggleShowLongErrorMessages
37	Nur IDL Formeln an/aus	Einstellung "Nur IDL Formeln" ändern	TogglePreciseFormulaEvaluation
IDL Zwischenablage v			
13	Zelle merken	Zwischenablage für Zell-Adressen füllen	ClipboardAddCurrentSelection
45	anzeigen	Zwischenablage für Zell-Adressen zeigen	ClipboardShow
14	löschen	Zwischenablage für Zell-Adressen löschen	ClipboardClear
Extras v			
Externe Verknüpfungen v			
42	prüfen	Externe Verknüpfungen prüfen	ExternalLinksCollectAndShow
43	anzeigen	Externe Verknüpfungen anzeigen	ExternalLinksShow
44	Prüfen beim Laden an/aus	Einstellung "Check Externe Verknüpfungen" ändern	ToggleCheckExternalLinks
IDL XLSLINK Hauptanwendung			
16	Status	IDL XLSLINK Hauptanwendung Statusinformation ausgeben	Status
15	Verbinden	IDL XLSLINK Hauptanwendung verbinden (NamedPipe - AddIn nach IDLOffice)	Verbinden
49	Anmeldung: IDL KONSIS	Anmeldung Datenbanken ausführen	PerformDBLogin
50	Informationen zu IDL XLSLINK	IDL System Info anzeigen	ShowIDLSystemInfo
Allgemeine Funktionen			
111	Get Current Status	Aktuellen Status abfragen (keine detaillierte Beschreibung im Code)	GetCurrentStatus
200	Block Update	XLSLink Update Funktion deaktivieren (keine detaillierte Beschreibung im Code)	BlockUpdate
201	Open Update	XLSLink Update Funktion aktivieren (keine detaillierte Beschreibung im Code)	OpenUpdate
Extras v			
Konvertierung v			
18	(Connector) abschließen	Konvertierung (Alt-Connector -> IDLOffice) abschließen (Nacharbeiten)	FinishConversion
36	(Connector) abschließen AddIn	Konvertierung (Alt-Connector -> IDLOffice) abschließen (Nacharbeiten) - innerhalb AddIn	FinishConversionFromAddIn
41	Format zurücksetzen	Konvertierung Zellenformatierung zurücksetzen	ResetCellFormat
Sonstige (nicht in Ribbon enthalten)			
10	Dokumentvariable bearbeiten	Dokumentvariable bearbeiten	DocVarEdit
11	Dokumentvariable löschen	Dokumentvariable löschen	DocVarDelete
12	Update Macros	Makros in aktueller Datei aktualisieren	UpdateMacros
31	RefreshRange Bezüge berücksichtigen	Beim RefreshRange Bezüge berücksichtigen	UpdateCellContentTypeBezug
32	RefreshRange IDLCombo berücksichtigen	Beim RefreshRange IDLCombo berücksichtigen	UpdateCellContentTypeCombo
34	Liste Konsis Exporte	Formular mit Konsis-Exporten zeigen	ShowExportJobList

Hints for VBA - Programmer/User

- The programming and use of macros is generally your own risk.
- We only provide you with the a way of using macros.
- The following VBA code lines are examples of how the various XLSLink functions can be called by macro.
- An individual adaptation to your environment or file structure will probably be necessary.
- The following VBA - code can be found at the end of this document as '.bas' - files, as well as a short instruction about the import into a VBA - project.
- Knowledge of VBA programming is required.
- We strongly encourage testing these macros in a test environment before using on production data
- The following two lines of code replace the old function call of the IDLConnector.
 - Both lines of code must always be used together.
 - „CallID“ is the corresponding number of the function according to the list above

```
Call Application.Run("IDLCall", CallID)  
Call WaitForReady
```
- The following requirements must always be met when calling functions.
 - A function must be correctly present uniquely in the respective VBA project.
 - A function checks the status / progress of the XLSLink and only returns to the procedure when the function has been fully executed.
 - You can adjust the seconds, although this is not strictly necessary.

```
Sub WaitForReady()
```

```
Dim tValue As Boolean
```

```
tValue = True
```

```
While tValue <> False
```

```
    VBA.DoEvents
```

```
    tValue = Application.Run("IDLCall", 111)
```

```
    If tValue <> False Then
```

```
        Application.Wait (Now + TimeValue("00:00:05"))
```

```
    End If
```

```
Wend
```

```
End Sub
```

The most important procedures for predefined XLSLink functions.

- You can copy these procedures straight into a VBA project.
- Procedure WaitForReady() must exist as a procedure in the VBA project.
- These procedures can either be called from an existing procedure.
- Or the call is made as an 'assigned macro' on a button in Excel.

- Group 'Aktualisieren':

```
Sub Aktualisieren_SelBereich()
```

```
Call Application.Run("IDLCall", 2)
```

```
Call WaitForReady
```

```
End Sub
```

```
Sub Aktualisieren_Tabellenblatt()
```

```
Call Application.Run("IDLCall", 3)
```

```
Call WaitForReady
```

```
End Sub
```

```
Sub Aktualisieren_Arbeitsmappe()
```

```
Call Application.Run("IDLCall", 4)
```

```
Call WaitForReady
```

```
End Sub
```

- Group 'Verpacken':

```
Sub Verpacken_SelBereich()
```

```
Call Application.Run("IDLCall", 19)
```

```
Call WaitForReady
```

```
End Sub
```

```
Sub Verpacken_Tabellenblatt ()
```

```
Call Application.Run("IDLCall", 20)
```

```
Call WaitForReady
```

```
End Sub
```

```
Sub Verpacken_Arbeitsmappe()
```

```
Call Application.Run("IDLCall", 21)
```

```
Call WaitForReady
```

```
End Sub
```

- Group 'Entpacken':

```
Sub Entpacken_SelBereich()
```

```
Call Application.Run("IDLCall", 22)
```

```
Call WaitForReady
```

```
End Sub
```

```
Sub Entpacken_Tabellenblatt ()
```

```
Call Application.Run("IDLCall", 23)
```

```
Call WaitForReady
```

```
End Sub
```

```
Sub Entpacken_Arbeitsmappe()
```

```
Call Application.Run("IDLCall", 24)
```

```
Call WaitForReady
```

```
End Sub
```

- Group 'Export':

- Mit diesen Prozeduren wird durch die Hauptanwendung ein Fenster mit eigener Steuerung aufgerufen.

- Ein Beispiel finden Sie unter der Nummer 3) auf Seite 8.

```
Sub Export_SelBereich()
```

```
Call Application.Run("IDLCall", 7)
```

```
Call WaitForReady
```

```
End Sub
```

```
Sub Export_Tabellenblatt ()
```

```
Call Application.Run("IDLCall", 8)
```

```
Call WaitForReady
```

```
End Sub
```

```
Sub Export_Arbeitsmappe()
```

```
Call Application.Run("IDLCall", 9)
```

```
Call WaitForReady
```

```
End Sub
```

Further procedures for predefined XLSLink functions.

- These procedures are not absolutely necessary.
- If major changes are made to a document via existing macros, it may be advisable to prevent all 'updates' automatically carried out by XLSLink using Block_Update().
- If the procedure Block_Update() is used, the procedure Open_Update() must be executed afterwards.
- Otherwise, no more IDL formulas will be calculated correctly.

```
Sub Block_Update()
```

```
Call Application.Run("IDLCall", 200)
```

```
Call WaitForReady
```

```
End Sub
```

```
Sub Open_Update()
```

```
Call Application.Run("IDLCall", 201)
```

```
Call WaitForReady
```

```
End Sub
```

- Sometimes it is necessary to control the correct execution of a procedure with a WAIT function.
 - This example waits for three seconds

```
Sub WAIT_Funktion()
```

```
Application.Wait (Now + TimeValue("00:00:03"))
```

```
End Sub
```

Example procedures for calling XLSLINK - Functions

- Here are examples of how several functions can be executed in sequence one after the other to give an idea of how the call can be programmed.
 - Adaptations to your existing documents will certainly have to be made.
 - These procedures given here can be written either as independent procedures, or as with calls to other procedures.
 - Procedure WaitForReady() must exist correctly as a procedure in the VBA project.
 - You will find these and other procedures in the attached file
- 1) The first example shows a dynamic call of function, here 'IDL Formula' - Editor.

```
Sub Dynamischer_IDLFunktionsaufruf()
```

```
Dim IDLCallID as Integer 'Variable definieren
```

```
IDLCallID = Tabelle1.Range("A1") 'Wert für Variable initialisieren (z.B. Zellinhalt 'A1'=1)
```

```
IDLFunktion_aufrufen IDLCallID 'Aufruf Subprozedur mit Übergabe der Variable
```

```
End Sub
```

```
Sub IDLFunktion_aufrufen (CallID As Integer) 'Funktionsaufruf mit übergebener Variablen
```

```
Call Application.Run("IDLCall", CallID)
```

```
Call WaitForReady
```

```
End Sub
```

- 2) This procedure updates a spreadsheet with packed formulas (Independent procedure).

```
Sub Tabellenblatt_Update()
```

```
Call Application.Run("IDLCall", 23) 'erst Tabellenblatt entpacken
```

```
Call WaitForReady
```

```
Call Application.Run("IDLCall", 3) 'danach Tabellenblatt aktualisieren
```

```
Call WaitForReady
```

```
Call Application.Run("IDLCall", 20) 'zu Letzt Tabellenblatt wieder verpacken
```

```
Call WaitForReady
```

```
End Sub
```


3) This procedure exports values of a selected range with packed formulas (procedure with call of further procedures).

- The export procedure calls a separate window, which requires a certain amount of time for initialisation.
- In further processing, selections must still be made and the export must be actively triggered.
- Unfortunately, this cannot be controlled by macros.
- A message box should be installed to ensure that the export process has been completed and the window closed.

Sub SelBereich_Export_SP() ' Voraussetzung ist ein selektierter Bereich

Entpacken_SelBereich 'erst Aufruf Subprozedur selektierten Bereich entpacken

Export_SelBereich 'danach Aufruf Subprozedur Werte aus selektiertem Bereich exportieren
MsgBox "Export beendet und Dialog geschlossen?" & Chr(13) & Chr(13) & "Wenn nicht, bitte warten und Dialog parallel mit 'Ende' schließen.", vbOKOnly, IDLFunktion

Verpacken_SelBereich 'zuletzt Aufruf Subprozedur selektierten Bereich wieder verpacken

End Sub

Sub Entpacken_SelBereich()

Call Application.Run("IDLCall", 22)

Call WaitForReady

End Sub

Sub Export_SelBereich()

Call Application.Run("IDLCall", 7)

Call WaitForReady

End Sub

Sub Verpacken_SelBereich()

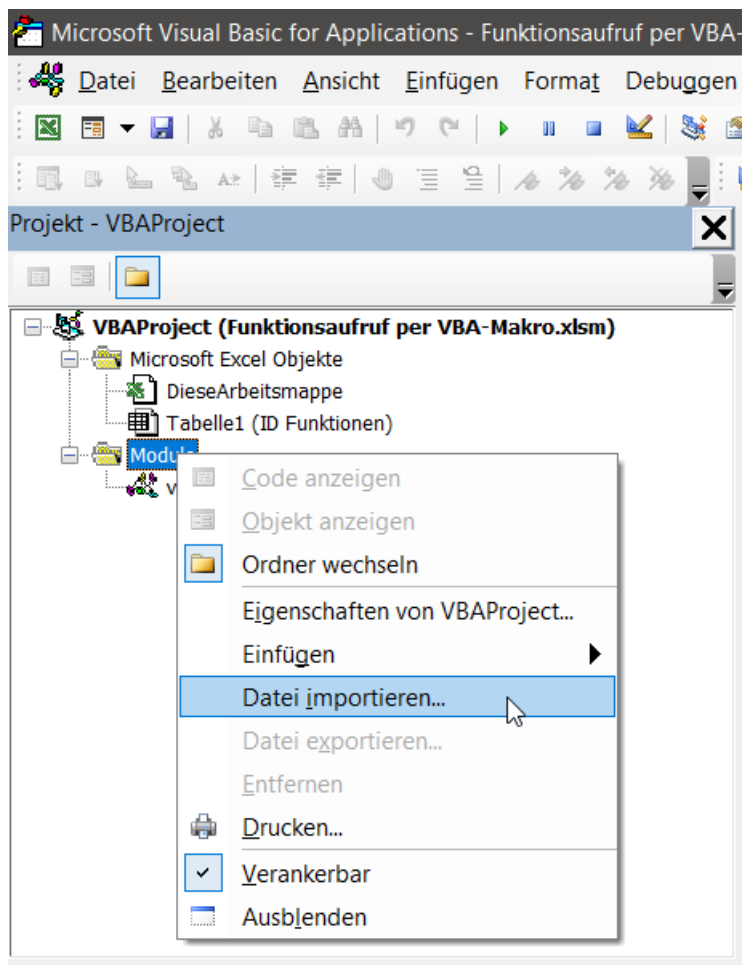
Call Application.Run("IDLCall", 19)

Call WaitForReady

End Sub

Import VBA - Code file into VBA - Project.

- Mark following code - files, copy them i.e. to „Desktop“
- Right click on Module and use menu 'import File' to add them to the current VBA project.



Complete VBA Code.

- Open these files with a suitable editor. Notepad ++' does a good job on this.



vbEINZELAKTION.bas



vbPROZEDUREN.bas