

Vorlage

SmartMaster-Vorlage

Eingaben

Ergebnisse

Arbeitsbereich

Variables

%Version1_C 2 : version counter
%Width #NAME? ; Monitor width in pixels
%Custom_Si #NAME? ; Custom size based on look
%CurrentSize 87 ; current size

	Width	CustomSize	
%Scrn_Size_	640	85	: Screen size lookup table
	800	95	
	1024	125	
	1152	140	
	1280	155	

#NAME?

%Pop_Looku? 1 A:E6
? 2 A:G6

%Lookup_CeA:E6
%Current_Po? 1

Arbeitsbereich

up into table

Texte

Bold items are referenced to duplicate s Translators, please ignore this table

	Translate this column	Xlat Name	Range Name	Range
	Amortization Sheet Strings	%CURRENTSIZ	CURRENTSIZ	B:B6..B:B6
<u>RangeNames</u>	<u>Strings</u>	%CURRENT_PC	CURRENT_PC	B:B22..B:B22
A_1	SmartMaster-Vorlage	%CUSTOM_SIZ	CUSTOM_SIZ	B:B5..B:B5
A_2		%LOOKUP_CE	LOOKUP_CE	B:B21..B:B21
A_3	Ergebnisse	%POP_LOOKU	POP_LOOKU	B:B17..B:C18
A_4	Eingaben	%SCRN_SIZE_T	SCRN_SIZE_T	B:B9..B:C13
		%VERSION1_CO	VERSION1_CO	B:B3..B:B3
		%WIDTH	WIDTH	B:B4..B:B4

Button Strings

RangeNames Strings

%B_1 Neue Version

Sheet names must be in order

Sheet Strings

RangeNames Strings

%S_1 Vorlage
 %S_2 Arbeitsbereich
 %S_3 Texte
 %S_4 Scripts

Function Strings

RangeNames Strings

%F_1 Bildschirm-Breite
 Do Not translate "Range Strings" starting with "%"

Range Strings

RangeNames Strings

%R_1 %Current_Poç Do Not Translate
 %R_2 Vorlage
 %R_3 %Width Do Not Translate
 %R_4 %Custom_Siz Do Not Translate
 %R_5 %CurrentSize Do Not Translate
 %R_6 %Scrn_Size_1 Do Not Translate
 %R_7 %Version1_Cç Do Not Translate
 %R_8 %Lookup_Cel Do Not Translate
 %R_10 %Amort_rangç Do Not Translate
 %R_11 Leer
 %R_12 Vorlage 1
 %R_13 Original
 %R_15 %Pop_Lookuç Do Not Translate

FileInfo Strings

RangeNames Strings

%D_1 SmartMaster Do Not Translate
 %D_2 Eigenen SmartMaster erstellen
 %D_3 Verwenden Sie diese Vorlage, um eigene SmartMaster-Schablonen zu erstellen.

Texte

Pop-up Strings

RangeNames Strings

%P_1 Doppelklicken Sie in diesen Textblock, und geben Sie Text ein, um einen Hilfetext zu e
%P_2 Wenn der Hilfetext fertig ist, "ffnen Sie den Script-Editor, und w,,hlen Sie das Objekt "?

Texte

```

# Ranges          8
Counter          9
Current Name     #NAME?
Xlat Name       #NAME?
Current Range    #NAME?

\R              {-- Create ranges based on new range names}
                {FOR K3;1;K2;1;K11}

Loop            {-- Create range name for current cell}    \U
                {CALC}
                /rnd@@(K4){calc}~
                /rnc@@(K5){calc}~@@(K6){calc}~           \W

```

RUN this macro before the one above

```

# Sheets          4
Counter          5
Current Sheet    #NAME?
Cur SheetName   Scripts
Current Range    %s_5

\S              {-- Renames all Sheets}
                {FOR K20;1;K19;1;K28}

Loop            {-- rename sheet}
                {CALC}
                #NAME?
                {IF K22=@@(K23)}{RETURN}
                {SHEET-NAME @@(K23);K22}

\X              {edit-copy d:b2..d:b200}

```

Pseudo-tranlator testing tool; pads all strings by %30 and adds INTL chars

```

\Y              {GOTO}Strings:b5~
\Q              {IF @cellpointer("row")>100}{QUIT}
                {IF @length(@cellpointer("contents"))<3}{Down}{BRANCH \
                {IF @cellpointer("type")="b"}{Down}{BRANCH \Q}
                #NAME?
                #NAME?
                {Down}{BRANCH \Q}

PAD            €□, f,,... †‡^%Š<Œ□Ž□□“”•——~™šœ¾¼P ¡¢£¤¥¦§¨_«¬€□,¡

```

Texte

```
Sub Translate
Dim X As Document
erstellen. Ziehen Sie den Text und das daz Dim X As Document
1" aus. L"schen Sie das Kommentarzeichen Set X = Bind( [],path + [].ShortName )
[Button 1].Text = [%B_1].CellValue

[Text 1].Text = [%P_1].CellValue
[Text 2].Text = [%P_2].CellValue
```

```
[%A_R1.Vorlage 1].Name = [%R_12].CellValue
[%A_R1.Leer].Name = [%R_11].CellValue
X.Subject = [%D_1].CellValue
X.Title = [%D_2].CellValue
X.Description = [%D_3].CellValue
X.NoDocInfo = FALSE
End Sub
```

Temp script and macro for conversion from .WK4 to .123

```
\Z      {-- Sets up For Gemini}
        {Create123}

Button script      Sub Create123
                   .NewButton 33200,30000,49800,34000
                   [Button 1].Text = [%B_1].CellValue

PopUp Script      [Picture 4].Name = "? 1"
                  [Picture 5].Name = "? 2"
                  .NewEditControl 11000,3500,85000,10500
                  [Text 1].Text = [%P_1].CellValue
                  [Text 1].Background.BackColor.ColorIndex = 2
                  [Text 1].EdgeLineWidth = $LineWidth2
                  [Text 1].TextHorizontalAlign = $AlignCenter
                  [Text 1].DesignerFrameStyle = $DesignerFrame2
                  .NewEditControl 55000,23200,95000,29000
                  [Text 2].Text = [%P_2].CellValue
                  [Text 2].Background.BackColor.ColorIndex = 2
                  [Text 2].EdgeLineWidth = $LineWidth2
                  [Text 2].TextHorizontalAlign = $AlignCenter
                  [Text 2].DesignerFrameStyle = $DesignerFrame2
```

Texte

CreateVersion Script	<pre>[%A_R1].NewVersion "Leer" [%A_R1].Clear [%A_R1].VersionBorderVisible = True [%A_R1.Original].Name = "Vorlage 1" [%A_R1.Vorlage 1].MakeCurrent</pre>
FileInformation Script	<pre>{}.Subject = [%D_1].CellValue {}.Title = [%D_2].CellValue {}.Description = [%D_3].CellValue {}.NoDocInfo = FALSE</pre>
Outline Script	<pre>[Vorlage].RowFolding = \$ParentBefore [Vorlage].RowOutlineVisible = \$Yes [A:A6..A:A10].DemoteRow 1 [A:A14..A:A27].DemoteRow 1 {}.PrintWhat = \$None End Sub</pre>

Translate all Strings in Strings Sheet.

Run macro \W to call Global Sub Translate, \S, \R and \X

Delete all Global Subs

Paste into IDE under Globals.

Cleanup file as you like

Rename from OS as *.12M

Test

Tool to unhide sheets

/whdb:a1..d:a1~

{Translate}

{S}

{R}

{X}

+ \Q}

f,,...††^%Š<Œ□Ž□□“”•——~™šœ¾⁄P ¡Œ£¤¥!\$”_«-€□,f,,...††^%Š<Œ□Ž□□“”•——~™šœ¾⁄P ¡Œ£¤¥!\$”_<

copy sub create123 to clip and paste to IDE
copy sub Translate to clip and paste to IDE
run macro \z
check all test blocks and buttons for looks
Set a title horizontal title from A:A5
Delete create123 script from IDE
Run Macro \X to copy scripts to Clip.
Paste into IDE under Globals.
Set up button, image and textblock scripts as per table below
hide and lock text blocks
Cleanup file
Hide scripts, strings and WorkArea sheets
test

Object	Event	Script
Picture 1	Selected	SM1
Picture 2	Selected	SM2
Picture 3	Selected	SM3
? 1,2, 3	Selected	Select_H
Button 1	Click	SM4
TextControl 1,	Selected	DeSelect_H

Texte

<-€□,f„...†‡^%Š<Œ□Ž□□““”•——™šœ¾⁄₄P ¡¢£¤¥¦§¨_<-€□,f„...†‡^%Š<Œ□Ž□□““”•——™šœ¾⁄₄P ¡¢£¤

Texte

¥|S" _<-€□, f,,... †‡^%Š<Œ□Ž□□"''''''•——™šœ¾4P iϕ£α¥|S" _<-€□, f,,... †‡^%Š<Œ□Ž□□"''''''•——™šœ¾4f

Texte

» iϕEα¥!S''_«-€□, f,,... †‡^%Š<Œ□Ž□□''''''•——~™šœ¾P iϕEα¥!S''_«-€□, f,,... †‡^%Š<Œ□Ž□□''''''•——~™

Scripts

Script Name	CODE	DESCRIPTION
Assign to Op	Sub Init	
Initialize		
or sub Init fo	[].Maximize	; need to add code here to cl
	End Sub	
Print	Sub SM1	
	[E5].Select	; select a cell to remove hanc
	.ShowAutomaticPageBreaks=False	;turn off the display of the pri
	.CurrentPrintSettings.PrintWhat = \$CurrentSheet	; Prints spreadsheet page
	.CurrentPrintSettings.FooterCenter = ""	
	.CurrentPrintSettings.Orientation = \$Portrait	: set print options
	.CurrentPrintSettings.FitToPage = \$FitRowsAndColumnns	
	.CurrentPrintSettings.LeftMargin = 1080	
	.CurrentPrintSettings.TopMargin = 1080	
	.CurrentPrintSettings.RightMargin = 1080	
	.CurrentPrintSettings.BottomMargin = 1080	
	[].ShowDrawLayer = False	
	[].ShowVersionBorders = False	; hide bitmaps
	CurrentApplication.Print	; turn off display of VM fames
	[].ShowDrawLayer = True	; Print
	[].ShowVersionBorders = True	; unhide bitmaps
	End Sub	; turn on display of VM fames
Help	Sub SM3	
	[E5].Select	; select a cell to remove hanc
	[].HelpContents	; raise help with right topic
	End Sub	
Create Versio	Sub SM4	
	DIM VMR AS Version	; dim VMS as Version
	[%A_R1].NewVersion [%R_2].CellValue + " " + Cstr([%A_R1].Clear	; Create new Version
	[%Version1_Count].Contents = CSTR([%Version1_Co	; Increment counter
	End Sub	
Send Results	Sub SM2	
	[E5].Select	; select a cell to remove hanc
	[].ShowDrawLayer = False	
	[].ShowVersionBorders = False	
	[].SelectAll	; copy results to clipboard
	[].CopyToClipboard \$BitMapFormat	
	[].SendMail , , "SmartMaster-Vorlage" , , , \$Clipboard	; send mail
	[].ShowDrawLayer = True	
	[].ShowVersionBorders = True	

Scripts

End Sub

```
TextBlock Sub Select_H
Scripts Dim X As String ; to hold popup number
Generic Select Dim A As EditText ;edit object
Dim CS as Range ; for range object for selectin
X = Right$( Cstr( .Name), 1) ; the number of the object

[%Current_Pop].Contents = "? " + X ; enter popup name into cell
[%Lookup_Cell].RecalcRange $Columns ; recalc cell
Set CS = Bind( [%Lookup_Cell].CellValue) ; bind cell ref to range object
CS.Select ; finally select the object

Set A = Bind ("Text " + X) ; set textcontrol object name
A.IsLocked = False ; unlock it either way
If A.Visible = True Then ; if visible
A.Visible = False ; make invisible

Else ; else
A.Visible = True ; make it visible

End IF
A.IsLocked = True
End Sub
```

```
Generic DeSeSub DeSelect_H If they click the TextBlock then deselect it and hide it.
Dim X As String
Dim A As EditText
Dim CS as Range
X = Right$( Cstr( .Name), 1)
Set A = Bind (.Name)

[%Current_Pop].Contents = "? " + X
[%Lookup_Cell].RecalcRange $Columns
Set CS = Bind( [%Lookup_Cell].CellValue)
CS.Select

A.IsLocked = False
A.Visible = False
A.IsLocked = True
End Sub
```


Scripts

neck for If Canada

bles from image
nt range

s

s

bles from image

bles from image

Scripts

Help Icons should be named ? 1; ? 2 etc.
Their corresponding lines, ellipses and text blocks will also
g cell