

Modello

Modello per SmartMasters

Immissioni

Risultati

WorkArea

Variables

%Version1_C 2 : version counter
%Width #NAME? ; Monitor width in pixels
%Custom_Size #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

WorkArea

up into table

Strings

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	Modello per SmartMasters	%CUSTOM_SIZ	CUSTOM_SIZ	B:B5..B:B5
A_2		%LOOKUP_CE	LOOKUP_CE	B:B21..B:B21
A_3	Risultati	%POP_LOOKU	POP_LOOKU	B:B17..B:C18
A_4	Immissioni	%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

<u>RangeNames</u>	<u>Strings</u>
%B_1	Nuova versione

Sheet names must be in order

Sheet Strings

<u>RangeNames</u>	<u>Strings</u>
%S_1	Modello
%S_2	WorkArea
%S_3	Strings
%S_4	Scripts

Function Strings

<u>RangeNames</u>	<u>Strings</u>
%F_1	larghezza_videata Do Not translate "Range Strings" starting with "%"

Range Strings

<u>RangeNames</u>	<u>Strings</u>
%R_1	%Current_PoçDo Not Translate
%R_2	Modello
%R_3	%Width Do Not Translate
%R_4	%Custom_SizDo Not Translate
%R_5	%CurrentSize Do Not Translate
%R_6	%Scrn_Size_1Do Not Translate
%R_7	%Version1_CçDo Not Translate
%R_8	%Lookup_CelDo Not Translate
%R_10	%Amort_rangçDo Not Translate
%R_11	Vuota
%R_12	Modello 1
%R_13	Originale
%R_15	%Pop_LookuçDo Not Translate

FileInfo Strings

<u>RangeNames</u>	<u>Strings</u>
%D_1	SmartMasters Do Not Translate
%D_2	Creazione SmartMaster personalizzato
%D_3	Modello base utilizzabile per creare i propri modelli SmartMaster.

Strings

Pop-up Strings

RangeNames Strings

%P_1 Per creare un popup di guida, fare doppio clic in questo blocco di testo e digitare il test

%P_2 Quando i popup sono pronti, attivare l'ambiente IDE e selezionare Object ? 1. Rimuove

Strings

# 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} {CALC} /rnd@@(K4){calc}~ /rnc@@(K5){calc}~@@(K6){calc}~	\U \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 ¡¢£¤¥¦§¨_«¬€□,¡

Strings

o. Trascinare quindi questo testo e l'icona
are il segno + per togliere il commento alla

```
Sub Translate
Dim X As Document
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.Modello 1].Name = [%R_12].CellValue
[%A_R1.Vuota].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
```

Strings

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

Strings

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£¤¥!\$”_<

Strings

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

Strings

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

Strings

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

Strings

» i€£¤¥¦§¨©ª«¬®¯°±²³´µ¶·¸¹º»¼½¾¿ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖ×ØÙÚÛÜÝÞßàáâãäåæçèéêëìíîïðññóôõö÷øùúûüýþÿ

Strings

šœ¼P jŁŁ¥!\$”_«¬€□,ƒ„…†‡^%Š<Œ□Ž□□“””•——™šœ¼P jŁŁ¥!\$”_«¬

Scripts

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

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