KSCL ToolBox
by Angus Wood - 100545,720
for Kingdom Software Consultants Ltd.

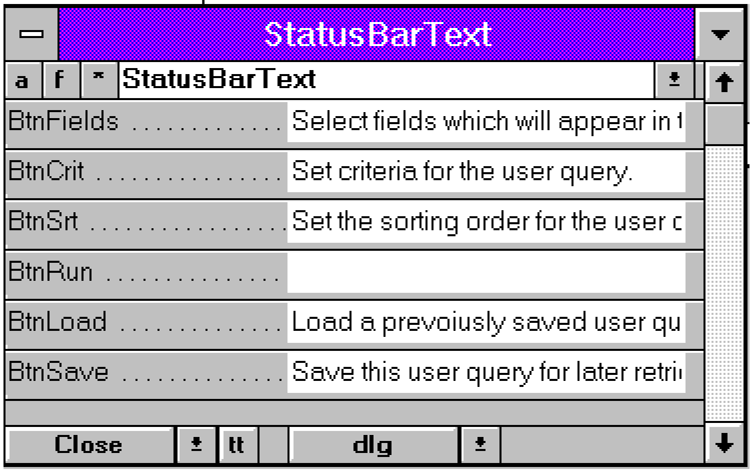
Property Side Slicer

Button Style Painter

Form Style Painter

About

Click for a description



StatusBarText

BtnFields	Select fields which will appear in t
BtnCrit	Set criteria for the user query.
BtnSrt	Set the sorting order for the user c
BtnRun	
BtnLoad	Load a prevoiusly saved user qu
BtnSave	Save this user query for later retri

Auto Grab

Select whether the toolbox updates itself every 2 seconds.
Default = Off

Toolbox will update itself when it receives the focus if this is not selected.

Freeze

Freeze/Unfreeze what is being displayed on the form.
Default = Off

Show All / Show Common

Select to show all properties in the combo-box.
Default = Off (common only)

Property Name

Pick the property to display/Edit.

Property Values

The values of the selected property for the highlighted controls.

Paint Button Style

Paint the highlighted command button with the style selected.

Select Button Style

Select the style to be loaded into the button painter.

Paint Form Style

Paint the target form with the attributes of the selected style.

Select Forms Style

Select the style to be loaded into the form painter.

ToolTips v3.0 Code Inserter

Inserts ToolTips v3.0 activation/deactivation code on to the selected controls.

About KSCL Toolbox

Created by **:** Angus Wood 100545,720 for
Kingdom Software Consultants Ltd.
21 Blair Street
Edinburgh
Tel : (+44 | 0) 131 220 4441
Fax : (+44 | 0) 131 220 4442

When **:** Mar 1995

Version **:** 1.2

Warranty **:** None - I'm not liable, that's the end of it. Use at your own risk.

Comments **:** If you find that it saves you time, take a moment to e-mail me with any comments you may have.

Properties Side Slicer

Button Style Painter

Using
Configuring

Form Style Painter

Using
Configuring

Using the Properties Side Slicer

Configuring

The properties side slicer allow you to individually view/edit a single property for a multiple selection, for example highlight all the command buttons on a form and edit their StatusBarText property without having to select/switch to properties sheet for each one.

Usage is simple, highlight the controls you wish to view/edit and the ToolBox will read your selection into a temp table then display the values in the main area of the form.

If applicable the values of the property will be displayed in a list box to provide faster selection, e.g. True/False for Visible.

As the values of the properties are edited the AfterUpdate event writes the value you entered back onto the target form.

Notes: The reader is not perfect, as I built the system my most important consideration was speed of operation and ease of use, because of this 'unreal' entries will appear, e.g. Caption for Embedded objects, If you edit these properties the write back routine will simply trap the error and discard the value.
While it would be easy to fix this, all the solutions I have come up with adversely affect the speed of operation and so were discarded.

Configuring the Properties Side Slicer

Using

The side slicer can be adjusted and extended to give a fair degree of flexibility, All the information is held in the table **z__TBX_PROP_VIEW**.

The table is laid out in the format;

prpType	prpName	Format	cboCols	cboValues
0	DisplayWhen		2	0;Always;1;Print Only;2;Screen Only
0	Width	General Number		
1	Caption			
0	RowSourceType		1	Table/Query;Value List;Field List

prpType (0 | 1) Is the property to be displayed when Show Common is selected.

prpName Text The name of the property to display/edit. Spelling must be exact.

cboCols Byte How many columns the property values combo box will have. Blank = no combo is displayed, edit box instead.

cboValues Text The RowSource for the property values combo, in ValueList format so semicolons are taken as row separators. BoundColumn is always zero, see the DisplayWhen record above.

Using the Button Style Painter

Configuring

Pick a style name from the combo box at the bottom left of the ToolBox, the button to the immediate left of the combo will display the name of the style.
Highlight the target button and press the bottom left button on the toolbox which be marked with the name of the button style.

If there are any non-command button controls highlighted these will be ignored.

If an event procedure already exists for a command button it will be left intact. Macros will be overwritten.

Configuring the Button Style Painter

Using

The button style painter is a compromise between speed and completeness, it does not attempt to make a ready to use command button but rather stores some basic attributes and allows you to paint them on to an existing command button.

Attributes are stored in the table **z__TBX_BUTTON_STYLES** in the format below.

BtnName	BtnCaption	BtnPic	BtnEventProc
Close	&Close	d:\pics\btns\close4.bmp	DoCmd Close A_FORM , "|"
Exit	E&xit	d:\pics\btns\exit4.bmp	Application.Quit

btnName Text Your name for the style.
btnCaption Text The caption for the button, ampersand accelerators as normal.
btnPic Text Full path and filename of any bitmap to be used, blank for none.
btnEventProc Text The event procedure to be assigned to the button, if you want to have the form name in the event proc use a pipe (| , chr$(124)) and it will be replaced at the time with the form's name.

Using the Form Style Painter

Configuring

The form style painter allows the storage and recall of virtually all form design parameters

Pick a style name from the combo box at the bottom right of the ToolBox, the button to the immediate left of the combo will display the name of the style. When pressed the button will set the parameters stored for that style on to the form.

Any invalid parameters will be ignored.

Configuring the Form Style Painter

Using

What it Can Do.

Set any **form** design parameter e.g. Width, HelpContextID,Caption

What if Cannot Do.

Set form **section** design parameters e.g. BackColor, Height.

How to configure it.

Attributes are stored in the table **z__TBX_FORM_STYLES** in the format below.

frmStyleName	Width	BorderStyle	DefaultView	ScrollBars
MenuFrm	9639	2	0	0
dlg	1200	3	0	0
RecStd	9639	2	0	1

Adding new styles

Styles are added as records in the z_TBX_FORM_STYLES table, the values used are those that you would use if stetting the property from Access Basic, in the example above the Width is in Twips. Search Access help on the property name and click the "Access Basic" pop-up to get the correct values.

Adding new parameters

Parameters are added by adding field definitions to the z_TBX_FORM_STYLES table with the name of the field corresponding to the name of the property you want to set.
For example if you wanted your styles to include whether the shortcut menu was available you would add a field called ShortCutMenu of type Yes/No to the table.
Spelling is vital, aside from that nothing else really matters as Access will twist the value to suit itself if the formatting is not correct.