OPENABAL

SING USER GUIDE

Abstract

This document describes the Open Abal version of Sing for LINUX, WINDOWS, MACOS.

Jamie Marshall

ijm@amenesik.com

Table of Contents

Introduction

This document provides an in-depth User Guide for SING, the OPEN ABAL Graphical User Interface design tool. All features described are equally available on LINUX, WINDOWS and MAC OS versions of OPENABAL. Any minor system differences will be indicated where appropriate.

Conventions

Throughout this document the following conventions will be respected:

- Examples in grey framed boxes represent information provided via dialog box input fields.

```
"filename"
```

- Examples in green framed boxes represent SING source code in WIDGET or FORMS methods.

```
MyTable.Collect()
For I = 1 to MyTable.response
MyTable.Locate(i)
Next i
```

- Examples in yellow framed boxes represent OPENABAL source code generated by SING to the output program.

```
Collect=handle,question:next,err,response,records
For x = 1 to response
  If ( x < response )
    primary = records(x)
    Search=handle,primary,/01:next,e,record,len(record)
  Else :: error = 103
 Endif
Next x
```

- Examples in blue framed boxes represent SING menu paths to launch actions.

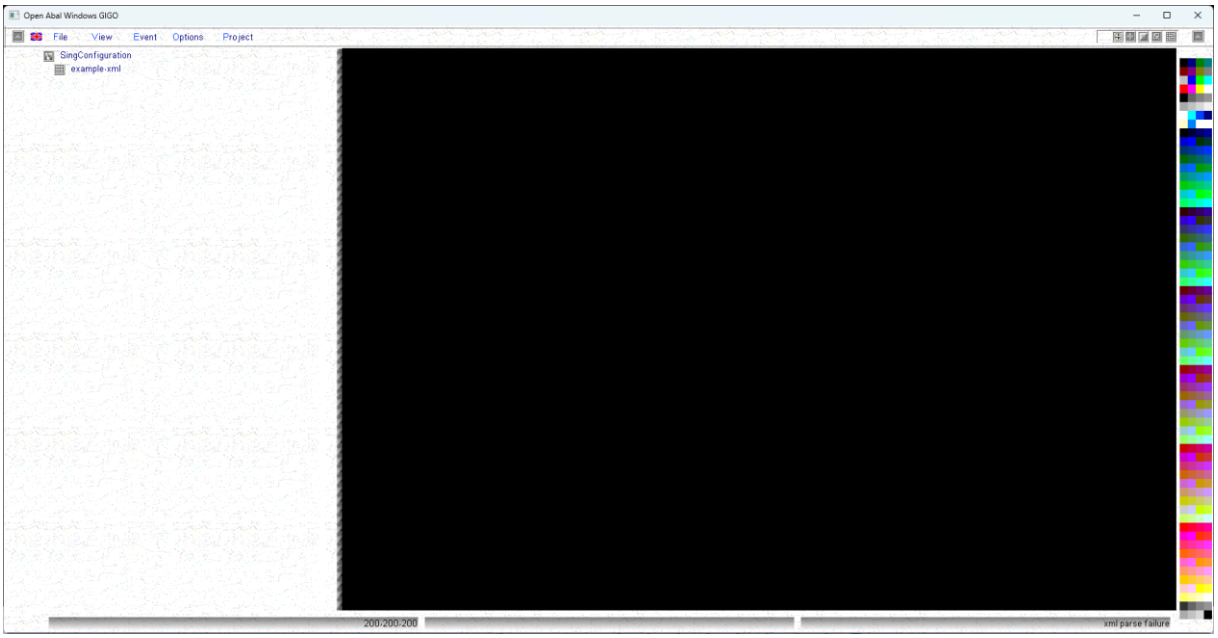
```
File. Save
```

- Examples in pink framed boxes represent SING Style Class instructions and classes.

```
example { text-color: red; border-style: solid; padding: 1mm; }
```

User Interface

The user interface of SING, as show below, comprises the following individual sections.



Top Menu Bar

The top menu bar, when visible, display the current contextual drop-down menu items.

The ESC key toggles visibility of the top menu and status bar.

Bottom Status Bar

The bottom status bar comprises three output zones displaying contextual status information.

The ESC key toggles visibility of the top menu and status bar.

Left Side Tool Bar

The contextual tool bar, when visible, provides access to the tool icons of the currently active editor occupying the central workbench.

Visibility of the contextual Tool Bar is controlled by the up/down control above it on the Top Menu bar.

Right Side Colour Palette

The color palette, when visible, allows quick selection of foreground and background colours.

Visibility of the contextual Colour Palette is controlled by the up/down control above it on the Top Menu bar.

Central Workbench

The central workbench is always visible and will be occupied by one of the seven tools made available with SING.

- Project (3D) Editor
- Visual Forms Editor
- Text Editor
- Image Editor
- Database Relations Model
- Component Flow Model
- Style Inheritance Model
- HTML Document Editor

Tree Viewer

The Tree Viewer, when visible, shows the structure of the currently loaded project(s).

The Tree Viewer visibility may be toggled using the View . Tree view menu option.

The width of the Tree View area may be changed by dragging the right-hand side edge to the left or right accordingly, using the LEFT button of the MOUSE.

Elements displayed in the tree view may be dragged onto the central work bench, using the LEFT button of the MOUSE, activating the corresponding Editor.

When multiple projects are loaded into SING the current project should be selected by dragging it into the workspace before its elements may be accessed.

A LEFT button MOUSE click will open an element (where appropriate) showing its nested collection of FORMS for PROJECT MODEL and WIDGETS for FORMS MODELS.

A RIGHT button MOUSE click on elements in the Tree View will open a contextual dialog box.

- **Project** (Sing Configuration) provides a dialog box allowing a component search by name to be performed.
- **Forms Model** provides access to the Forms Properties dialog box of the corresponding visual form.
- **Forms Widgets** provides access to the Widget Properties dialog box of the corresponding widget.

Forms Widgets may be dragged and dropped, to after a targeted widget of the same FORMS MODEL, using the LEFT button of the MOUSE.

National Language

SING supports managing up to eight national language message and help texts for both the tool and any visual dialog forms created with it.

The current National Language may be changed using the left and right buttons of the mouse on the national language flag displayed to the left of the Top Menu bar.

Quick Access Tool Tray

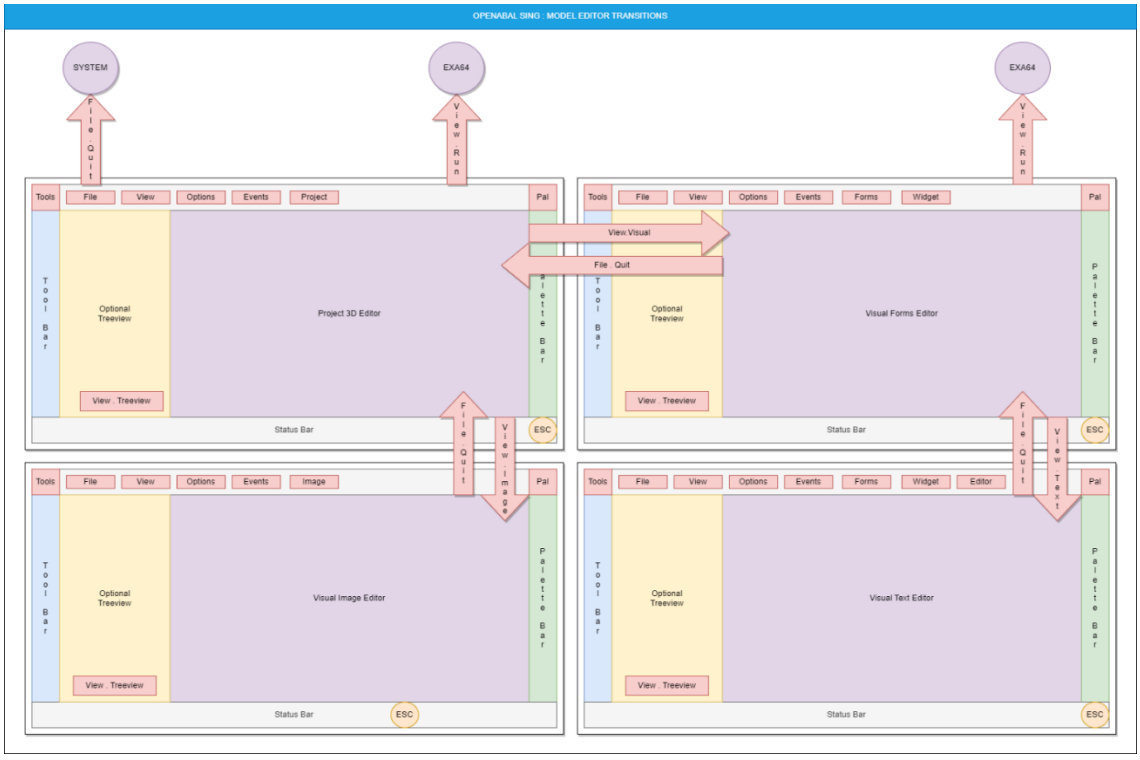
The quick access contextual tool tray, to the right of the Top Menu bar, provides one click access to the most frequently used operations.

- Lock / Unlock of the project and all components.
- Launch Debugger for the currently selected component.
- Launch Runtime for the currently selected component.
- Launch Text Editor for the currently selected component.
- Launch Translation for the currently selected component.
- Launch Linker for the currently selected component.

Where appropriate, the LEFT button of the MOUSE will launch the tool, whilst the RIGHT button will provide access to the configuration dialog box of the corresponding tool.

Editor Transition Summary

The following diagram depicts the primary transitions between the different model editors.

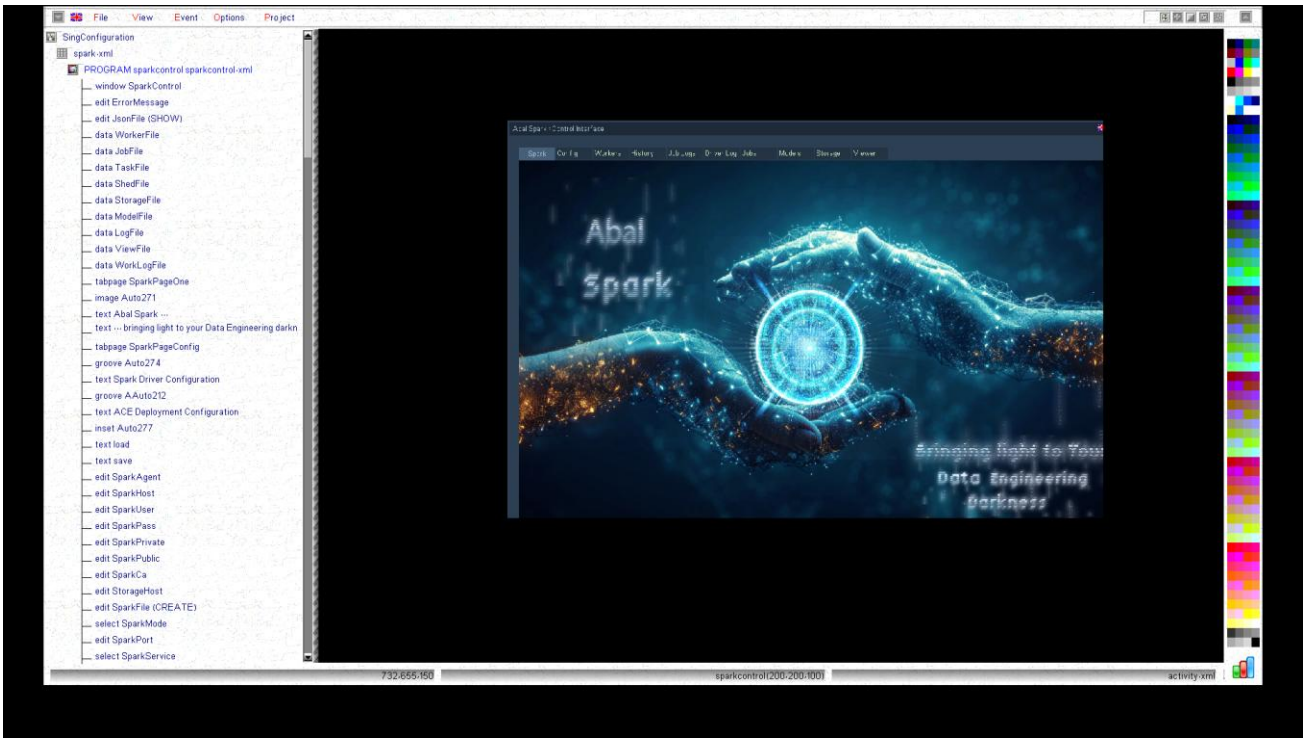


PROJECT (3D) EDITOR

The PROJECT or APPLICATION editor is a 3D workspace allowing access to all images, source files, and forms models attached to your Application Project.

All elements are sequentially chained on the PROJECT model list and are visible at their 3D coordinate in the 3D viewer.

Elements that are nearer, to the observer, in the direction of vision, will cover or hide elements that are further away.



NAVIGATION

The 3D viewer of the PROJECT / APPLICATION workbench may be navigated using the keyboard as follows:

- DOWN ARROW (CTRL E) selects the next sequential element in the list.
- UP ARROW (CTRL K) selects the pervious sequential element in the list.
- LEFT ARROW (CTRL H) activates the Keyboard Help dialog box.
- RIGHT ARROW (CTRL F) activates the Component Flow Model editor.
- TABULATION (CTRL I) opens the 3D object properties dialog box for inspection and modification of the position and dimensions of the visual element.
- ENTER (CTRL M) launches the contextual editor in accordance with the element type.
- PAGE DOWN (CTRL C) scrolls the 3D viewer downwards.
- PAGE UP (CTRL R) scrolls the 3D viewer upwards.
- END (CTRL B) positions to the last element in the list.

- DELETE deletes the current element from the project model.
- CTRL N pulls the current element and all subsequent elements downwards in the 3D viewer.
- CTRL P opens the Production Parameters dialog box.
- CTRL Y pulls the current element and all subsequent elements upwards in the 3D viewer.
- CTRL Z opens the Database Model Editor.
- 'F' changes the 3D camera view to forwards (default) looking towards the origin at coordinates [X=0, Y=0, Z=0].
- 'B' changes the 3D camera view to backwards looking away from the origin towards coordinates [X=0, Y=0, Z=INF].
- 'L' changes the 3D camera view to look leftwards towards the X origin coordinates [X=0, Y=0, Z=?].
- 'R' changes the 3D camera view to look rightwards towards coordinates [X=INF, Y=0, Z=?].
- 'U' changes the 3D camera view to look upwards towards the Y origin coordinates [X=?, Y=0, Z=?].
- 'D' changes the 3D camera view to look downwards towards coordinates [X=?, Y=INF, Z=?].

The 3D viewer of the PROJECT / APPLICATION workbench may be navigated using the MOUSE as follows:

- SCROLL WHEEL UP displacement along the Z AXIS, in or out, depending on the current viewing direction (F, or B)
- SCROLL WHEEL DOWN displacement along the Z AXIS, in or out, depending on the current viewing direction (F, or B)
- LEFT BUTTON DRAG, in empty space, to the left or right, will move the 3D model view accordingly.
- LEFT BUTTON DRAG, in empty space, upwards or downwards, will move the 3D model view accordingly.
- LEFT BUTTON DRAG, on an element in the 3D viewer, to the left or right, will move the element accordingly.
- LEFT BUTTON DRAG, on an element in the 3D viewer, upwards or downwards, will move the element accordingly.
- RIGHT BUTTON on an element in the 3D viewer will display a help bubble containing the objects properties.
- LEFT BUTTON PRESS + RIGHT BUTTON CLICK will open the contextual editor of the selected element of the model.

MENUS

The Top Menus, and options, that are accessible when the Project or Application editor is active, are described here.

File Menu

This menu allows access to options relating to the PROJECT or APPLICATION file.

New

This menu option will reset the application project to empty after first checking if any pending modifications are to be saved.

Save

This menu option saves the state of the current project to its current XML file on storage.

Save As

This menu option opens a dialog box allowing the state of the current project to be saved to an alternative project name.

Open

This menu option opens a dialog box for selection of an application project to be loaded into the Tree View alongside the current project. Multiple projects can be loaded in this way.

Close

This menu option closes all loaded projects and quits the Open Abal Graphical Design tool.

Production

This menu option opens the Project Production Parameters dialog box allowing Project level production operations to be launched.

Compare

This menu option opens the File Compare dialog box allowing two files to be selected for line by line compare and visualisation of their respective differences.

About

This menu option displays the information dialog frame showing the current version of the Open Abal Graphical Design tool.

Quit

This menu option closes all loaded projects and quits the Open Abal Graphical Design tool.

View Menu

This menu allows access to various editors and dialog boxes of the Open Abal Graphical Design tool.

Run

This menu option allows access to the Runtime / Debug Properties dialog box for launch or debug of Open Abal programs

Text

This menu option allows access to the Text Editor for the current form or source.

Image

This menu option allows access to the Image Editor for the inspection and modification of image elements or the representation of a visual form elements.

Font

This menu option provides access to the Font Editor dialog box allowing inspection and modification of graphical text fonts.

Properties

This menu option allows access to the 3D Object Properties dialog box of the currently selected element of the 3D model.

Visual

This menu option allows access to the Visual Forms Editor for the currently selected visual forms element of the 3D Model.

Database

This menu option provides access to the Database Editor dialog box allowing inspection, creation and modification of database table descriptions.

Tree View

This menu option toggles the display of the Tree View.

Application

This menu option allows return to the 3D Model view of the Project or Application Editor, from the Text Editor, Visual Forms Editor, Image Editor or the Database, Flow and Style Model Editors.

Style

This menu option provides access to the Style Manager and Editor dialog box allowing inspection, creation and modification of visual style class descriptions.

Model

This menu option displays the secondary Model Menu allowing access to the:

- Database Relations Model Editor
- Program Flow Model Editor
- Style Class Model Editor

Event Menu

This Top Menu option displays the event menu options, most of which serve no purpose for the Project or Application Editor.

OnCreate

This menu option serves no purpose in the 3D Model Project Editor.

OnRemove

This menu option serves no purpose in the 3D Model Project Editor.

OnShow

This menu option serves no purpose in the 3D Model Project Editor.

OnHide

This menu option serves no purpose in the 3D Model Project Editor.

OnFocus

This menu option serves no purpose in the 3D Model Project Editor.

OnLose

This menu option serves no purpose in the 3D Model Project Editor.

OnEvent

This menu option serves no purpose in the 3D Model Project Editor.

Item Document

This menu option serves no purpose in the 3D Model Project Editor.

Form Document

This menu option allows access to the HTML Document Editor for the inspection and modification of the Online Help document for the Project or Application.

Options

This Top Menu option gives access to the different configuration dialog boxes.

Parameters

This menu option allows access to the centralised General Parameters dialog box.

Pragmas

This menu option allows access to the centralised Translator Pragmas dialog box.

Configure

This menu option allows access to the centralised SING Configuration dialog box.

Screen Capture

This menu option allows access to the centralised Screen Capture dialog box.

Animate

This menu option allows access to the centralised Project Animator dialog box.

Hypertext

This menu option allows access to the centralised HTML Document Parameters dialog box.

Tree View

This menu option allows access to the centralised Tree View Parameters dialog box.

Project

This option of the Top Menu allows access to the menu options specific to the Project or Application 3D Model editor.

Camera

This menu option provides access to the 3D Model Camera Properties Global Project Configuration dialog box allowing configuration of the 3D model editor.

Translate

This menu option provides access to the National Language Translation Options dialog box allowing cross translation of identical national language texts across the collection of visual forms comprising an Application / Project model.

Reorganise

This menu option provides access to the Sort Camera dialog box allowing visual and 3D special reorganisation of the 3D Objects of the Project / Application Model.

Help Text

This menu option provides access to the Help Text International Document Editor.

Add Form

This menu option provides access to the 3D Object Properties dialog box allowing selection of addition of a NEW Visual Form component to the Project / Application Model. The new element will be added after the currently select element.

Add Source

This menu option provides access to the 3D Object Properties dialog box allowing selection of addition of a NEW Textual Source File component to the Project / Application Model. The new element will be added after the currently select element.

Add Image

This menu option provides access to the 3D Object Properties dialog box allowing selection of addition of a NEW Image File component to the Project / Application Model. The new element will be added after the currently select element.

Export Labels

This menu option provides access to the Export Labels Parameters dialog box allowing national language labels and help texts to be exported for manual cross translation.

Import Labels

This menu option provides access to the Import Labels Parameters dialog box allowing national language labels and help texts to be imported after manual cross translation.

Sort by Name

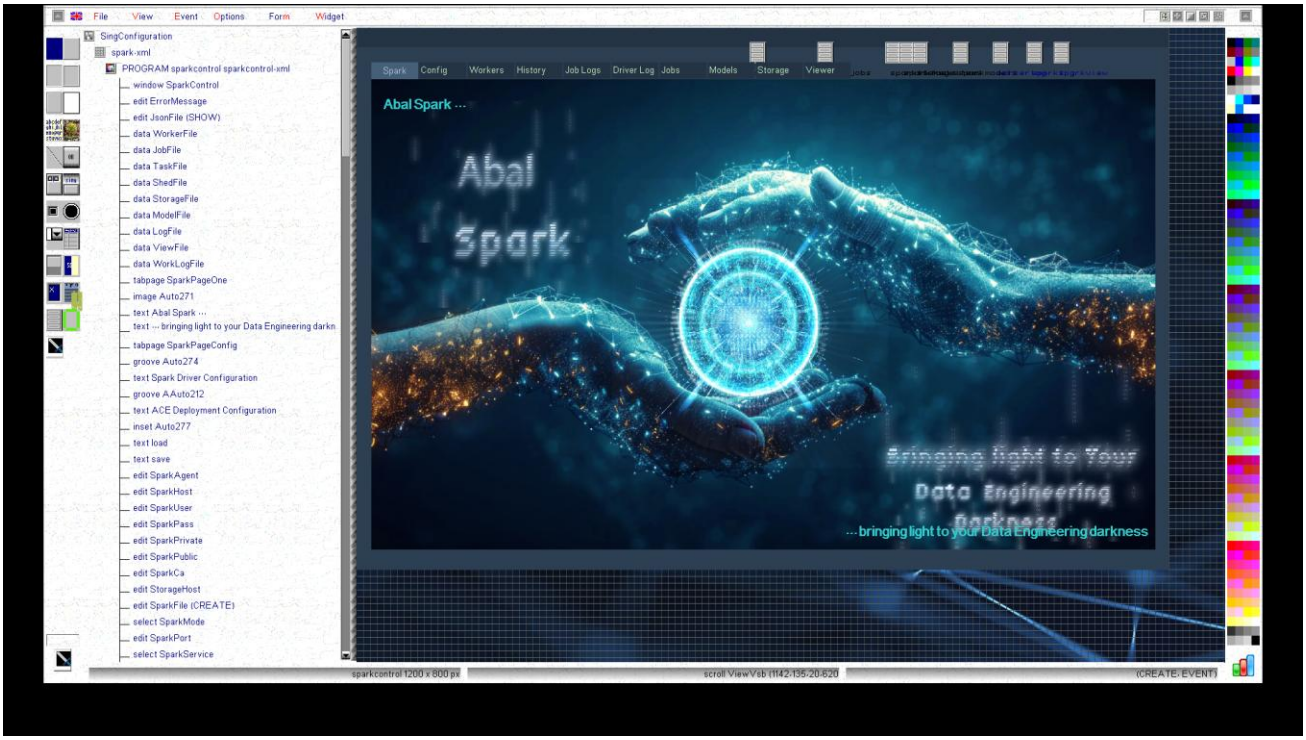
This menu option sorts the 3D Objects of the Project / Application Model by name.

Sort by Type

This menu option sorts the 3D Objects of the Project / Application Model by type.

VISUAL FORMS EDITOR

The Visual Forms Editor provides a collection of tools and menu functions facilitating the design and implementation of graphical user interfaces for Open Abal programs.



Visual forms comprise:

- The Forms properties describing global parameters of the form.
- A sequential collection of Forms Objects also referred to as Widgets.
- A collection of Forms Methods describing the required behaviour of the Form.
- A collection of Forms Members providing supplementary storage variables.

NAVIGATION

The Forms Editor may be navigated using the keyboard as follows:

- DOWN ARROW (CTRL E) selects the next sequential element in the Forms widget list.
- UP ARROW (CTRL K) selects the pervious sequential element in the Forms widget list.
- LEFT ARROW (CTRL H) activates the Keyboard Help dialog box.
- RIGHT ARROW (CTRL F) activates the Component Flow Model editor.
- TABULATION (CTRL I) opens the Widget Properties dialog box for the currently selected Widget.
- ENTER (CTRL M) opens the Forms Properties dialog box.
- ESCAPE Toggle display of the Menu and Status Bars.
- PAGE DOWN (CTRL C) selects and displays the next TAB PAGE element.
- PAGE UP (CTRL R) selects and displays the previous TAB PAGE element.

- DELETE deletes the currently selected widget.
- CTRL V toggles between styled and standard display modes.
- CTRL G toggles between GRID LOCKED and FREE FORM modes.
- CTRL Z displays the Database Relations Model Editor.
- CTRL O inserts a new Widget after the currently selected widget s directed by the currently selected Forms Model Tool.
- “T” Toggles the state of the Tree View.”
- “E” Launch the program execution.
- “S” Opens the Forms Model Save dialog box.
- “Q” launches the quick save of the Forms Model.
- “P” Launch the Forms Model production.

The Forms Editor may be navigated using the MOUSE:

- LEFT BUTTON CLICK selection of the current widget.
- LEFT BUTTON DRAG inside a widgets zone repositions the selected widget until MOUSE UP.
- LEFT BUTTON DRAG on a widget’s edge re-dimensions the selected widget until MOUSE UP.

TOOLS

The Widgets of a Visual Forms Model can be created using one of the following Widget Tools provided by the Forms Editor.

Fill Zone

This tool creates a decorative Fill Widget using the provided position, dimensions, colours and options.

Inset Frame

This tool creates an INSET Frame Widget using the provided position, dimensions and the standard GUI relief colours.

Outset Frame

This tool creates an OUTSET Frame Widget using the provided position, dimensions and the standard GUI relief colours.

Ridge Frame

This tool creates q RIDGE Frame Widget using the provided position, dimensions and the standard GUI relief colours.

Groove Frame

This tool creates a GROOVE Frame Widget using the provided position, dimensions and the standard GUI relief colours.

Edit Frame

This tool creates an EDIT Frame Widget using the provided position, dimensions and standard GUI relief colours.

Draw Text

This tool creates a TEXT Drawing Widget using the provided position, dimensions, text font, foreground and background colours, message string and text alignment options.

Draw Image

This tool creates an IMAGE Drawing Widget using the provided position, dimensions, image filename, foreground and background colours, message string and image alignment options.

Draw Line

This tool creates a LINE Drawing Widget using the provided start and end positions, and foreground colour.

Push Button

This tool creates a textual of image PUSH BUTTON Drawing Widget using the provided position, dimensions, image or message, foreground and background colours, image alignment options.

Window

This tool creates a WINDOW Drawing Widget using the provided position, dimensions, title message and supplementary windows options.

Tab Page

This tool creates a TAB PAGE Button and Frame Drawing Widget using the provided position, dimensions, title message and supplementary options.

Check Box

This tool creates a CHECK BOX Drawing Widget using the provided position, dimensions, label, foreground and background colours, and alignment options.

Radio Button

This tool creates a RADIO BUTTON Drawing Widget using the provided position, dimensions, label, foreground and background colours, and alignment options.

Selection List

This tool creates a SELECTION LIST Drawing Widget using the provided position, dimensions, messages, foreground and background colours, and alignment options.

Data Table

This tool creates a DATA TABULATOR Drawing Widget using the provided position, dimensions, titles, formats and data messages, foreground and background colours, and alignment options.

Scroll Bar

This tool creates a vertical or horizontal SCROLL BAR Drawing Widget using the provided position, dimensions, foreground and background colours, and alignment options.

Progress Bar

This tool creates a horizontal PROGRESS BAR Drawing Widget using the provided position, dimensions, labels, foreground and background colours, and alignment options.

Switch Button

This tool creates a SWITCH BUTTON Drawing Widget using the provided position, dimensions, labels, foreground and background colours, and alignment options

Data Graph

This tool creates a DATA GRAPH Drawing Widget using the provided position, dimensions, foreground and background colours, alignment options and data values.

Data Object

This tool creates a complex DATA OBJECT using the provided position, dimensions, and type specific parameters.

Widget Selector

This tool allows definition of a selection of widgets to be acted upon by certain Widget Menu operations.

Widget Pointer

This tool allows selection of individual Widgets to be inspected, repositioned or re-dimensioned.

MENUS

File

This menu allows access to options relating to the PROJECT or APPLICATION file.

New

This menu option will reset the current Visual Form to empty after first checking if any pending modifications are to be saved.

Save

This menu option saves the state of the current Visual Form to its current XML file on storage.

Save As

This menu option opens a dialog box allowing the state of the current Visual Form to be saved to an alternative Visual Form name.

Open

This menu option opens a dialog box for selection of a Visual Form to be loaded into the Visual Forms Editor replacing the current Visual Form.

Close

This menu option serves no purpose in the Visual Form Editor.

Production

This menu option opens the Forms Production Parameters dialog box allowing current Visual Form production operations to be launched.

Compare

This menu option opens the File Compare dialog box allowing two files to be selected for line by line compare and visualisation of their respective differences.

About

This menu option displays the information dialog frame showing the current version of the Open Abal Graphical Design tool.

Quit

This menu option closes all loaded projects and quits the Open Abal Graphical Design tool.

View Menu

This menu allows access to various editors and dialog boxes of the Open Abal Graphical Design tool.

Run

This menu option allows access to the Runtime / Debug Properties dialog box for launch or debug of Open Abal programs

Text

This menu option allows access to the Text Editor for the current form or source.

Image

This menu option allows access to the Image Editor for the inspection and modification of image elements or the representation of a visual form elements.

Font

This menu option provides access to the Font Editor dialog box allowing inspection and modification of graphical text fonts.

Properties

This menu option allows access to the 3D Object Properties dialog box of the currently selected element of the 3D model.

Visual

This menu option allows access to the Visual Forms Editor for the currently selected visual forms element of the 3D Model.

Database

This menu option provides access to the Database Editor dialog box allowing inspection, creation and modification of database table descriptions.

Tree View

This menu option toggles the display of the Tree View.

Application

This menu option allows return to the 3D Model view of the Project or Application Editor, from the Text Editor, Visual Forms Editor, Image Editor or the Database, Flow and Style Model Editors.

Style

This menu option provides access to the Style Manager and Editor dialog box allowing inspection, creation and modification of visual style class descriptions.

Model

This menu option displays the secondary Model Menu allowing access to the:

- Database Relations Model Editor
- Program Flow Model Editor
- Style Class Model Editor

Event Menu

This Top Menu option displays the event menu options allowing access to the explicit event method text defined for the currently selected Widget of the Visual Form.

OnCreate

This menu option provides direct access to the Text Editor for the method text of the OnCreate event of currently selected Widget of the currently loaded Visual Forms Model.

The OnCreate method of a widget will be invoked implicitly from the OnCreate method of the parent Visual Form **after** all implicit construction required by the specific widget type has been performed.

OnCreate methods may be invoked explicitly from any Event Method text of any Widgets of the same Visual Form, or from any Forms Method text of the parent Visual Form.

OnRemove

This menu option provides direct access to the Text Editor for the method text of the OnRemove event of currently selected Widget of the currently loaded Visual Forms Model.

The OnRemove method of a widget will be invoked implicitly from the OnRemove method of the parent Visual Form **before** any implicit destruction required by the specific widget type has been performed.

OnRemove methods may also be invoked explicitly from any Event Method text of any Widgets of the same Visual Form, or from any Forms Method text of the parent Visual Form.

OnShow

This menu option provides direct access to the Text Editor for the method text of the OnShow event of currently selected Widget of the currently loaded Visual Forms Model.

The OnShow method of a widget will be invoked implicitly from the OnShow method of the parent Visual Form **in place** of any implicit **display** operations required for the specific widget type.

OnShow methods may also be invoked explicitly from any Event Method text of any Widgets of the same Visual Form, or from any Forms Method text of the parent Visual Form.

OnHide

This menu option provides direct access to the Text Editor for the method text of the OnHide event of currently selected Widget of the currently loaded Visual Forms Model.

The OnHide method of a widget will be invoked implicitly from the OnHide method of the parent Visual Form **in place** of any implicit **disappearance** operations required for the specific widget type.

OnHide methods may also be invoked explicitly from any Event Method text of any Widgets of the same Visual Form, or from any Forms Method text of the parent Visual Form.

OnFocus

This menu option provides direct access to the Text Editor for the method text of the OnFocus event of currently selected Widget of the currently loaded Visual Forms Model.

The OnFocus method of a widget will be invoked implicitly from the OnFocus method of the parent Visual Form **in place** of any implicit **focus management** operations performed for the specific widget type.

OnFocus methods may also be invoked explicitly from any Event Method text of any Widgets of the same Visual Form, or from any Forms Method text of the parent Visual Form.

OnLose

This menu option provides direct access to the Text Editor for the method text of the OnLose event of currently selected Widget of the currently loaded Visual Forms Model.

The OnLose method of a widget will be invoked implicitly from the OnLose method of the parent Visual Form **in place** of any implicit **focus release** operations performed for the specific widget type.

OnLose methods may also be invoked explicitly from any Event Method text of any Widgets of the same Visual Form, or from any Forms Method text of the parent Visual Form.

OnEvent

This menu option provides direct access to the Text Editor for the method text of the OnEvent event of currently selected Widget of the currently loaded Visual Forms Model.

The OnEvent method of a widget will be invoked implicitly from the OnEvent method of the parent Visual Form **in place** of any implicit **event processing** operations performed for the specific widget type.

OnEvent methods may also be invoked explicitly from any Event Method text of any Widgets of the same Visual Form, or from any Forms Method text of the parent Visual Form.

Item Document

This menu option allows access to the HTML Document Editor for the inspection and modification of the Online Help document for the currently selected Widget of the Visual Form.

Form Document

This menu option allows access to the HTML Document Editor for the inspection and modification of the Online Help document for the current Visual Form.

Options

This Top Menu option is identical to the that of the Project / Application Editor and gives access to the different configuration dialog boxes.

Parameters

This menu option allows access to the centralised General Parameters dialog box.

Pragmas

This menu option allows access to the centralised Translator Pragmas dialog box.

Configure

This menu option allows access to the centralised SING Configuration dialog box.

Screen Capture

This menu option allows access to the centralised Screen Capture dialog box.

Animate

This menu option allows access to the centralised Project Animator dialog box.

Html

This menu option allows access to the centralised HTML Document Parameters dialog box.

Tree View

This menu option allows access to the centralised Tree View Parameters dialog box.

Form Menu

This Top Menu option provides access to all Visual Forms specific menu options.

Properties

This menu option provides access to the Form Properties dialog box allowing inspection and modification of the form properties of this Visual Form.

Methods

This menu option provides access to the Forms Method Quick Access dialog box of Forms Methods declared for the Visual Form.

Document

This menu option provides access to the Draft Document Page Parameters.

Help Text

This menu option provides access to the International Help Text editor dialog box.

Re Numerate

This menu option performs renumeration of the unnamed widgets of the form to ensure their unicity.

Triggers

This menu option recalculates all Widget Trigger codes to ensure unicity if this is possible

Interface

This menu option generates the Visual Forms interface.

Documentation

This menu option provides access to the Html Document Production Parameters dialog box for the current Visual Form.

Full Screen

This menu option toggles the display of Forms Models larger than the screen as either clipped forms or best fit forms.

Style

This menu option provides access to the Text Editor for the style sheet defined for the current Visual Form.

Undo

This menu option rolls back the last change to the current Visual Form.

Redo

This menu option rolls forward the last undone change to the current Visual Form.

Compare

This menu option provides access to the Compare Forms dialog box allowing two Visual Form Model files to be compared.

Import Abal Source

This menu option provides access to the Abal Source Import Operation dialog box allowing an Abal Source to be imported collecting variables and procedures as Form Members and Methods.

Widget Menu

This Top Menu option provides access to all Widget specific menu options.

Messages

This menu option provides access to the International Message Editor dialog box for the currently selected widget.

Help

This menu option provides access to the International Document Editor dialog box for the currently selected widget

Font

This menu option sets the value of the Font Property of the current widgets using the value taken from the **Options Parameters Font** property.

Type

This menu option sets the value of the Type Property of the current widgets using the value taken from the **Options Parameters Type** property.

Align

This menu option sets the value of the Align Property of the current widgets using the value taken from the **Options Parameters Align** property.

Bold

This menu option sets the value of the Bold Property of the current widgets using the value taken from the **Options Parameters Bold** property.

Shadow

This menu option sets the value of the Shadow of the current widgets using the value taken from the **Options Parameters Shadow** property.

Underline

This menu option sets the value of the Underline of the current widgets using the value taken from the **Options Parameters Underline** property.

Copy

This menu option copies the currently selected widgets **to** the Visual Form Editor clipboard.

Paste

This menu option pastes the currently selected widgets **from** the Visual Form Editor clipboard.

Lock

This menu option locks the currently selected widgets.

Unlock

This menu option unlocks the currently selected widgets.

Automatic Display

This menu option sets the value of the Automatic Display Property for the currently selected widgets.

Conditional Show

This menu option sets the value of the Conditional Show Property for the currently selected widgets.

Foreground Colour

This menu option sets the value of the Foreground Colour of the currently selected widgets from **Palette Foreground** value.

Background Colour

This menu option sets the value of the Background Colour of the currently selected widgets **Palette Background** value.

Locate

This menu option provides access to the Quick Access list of usages of the currently selected widget.

Style

This menu option provides access to the Style selector allowing selection of the style of the currently selected widgets

Delete

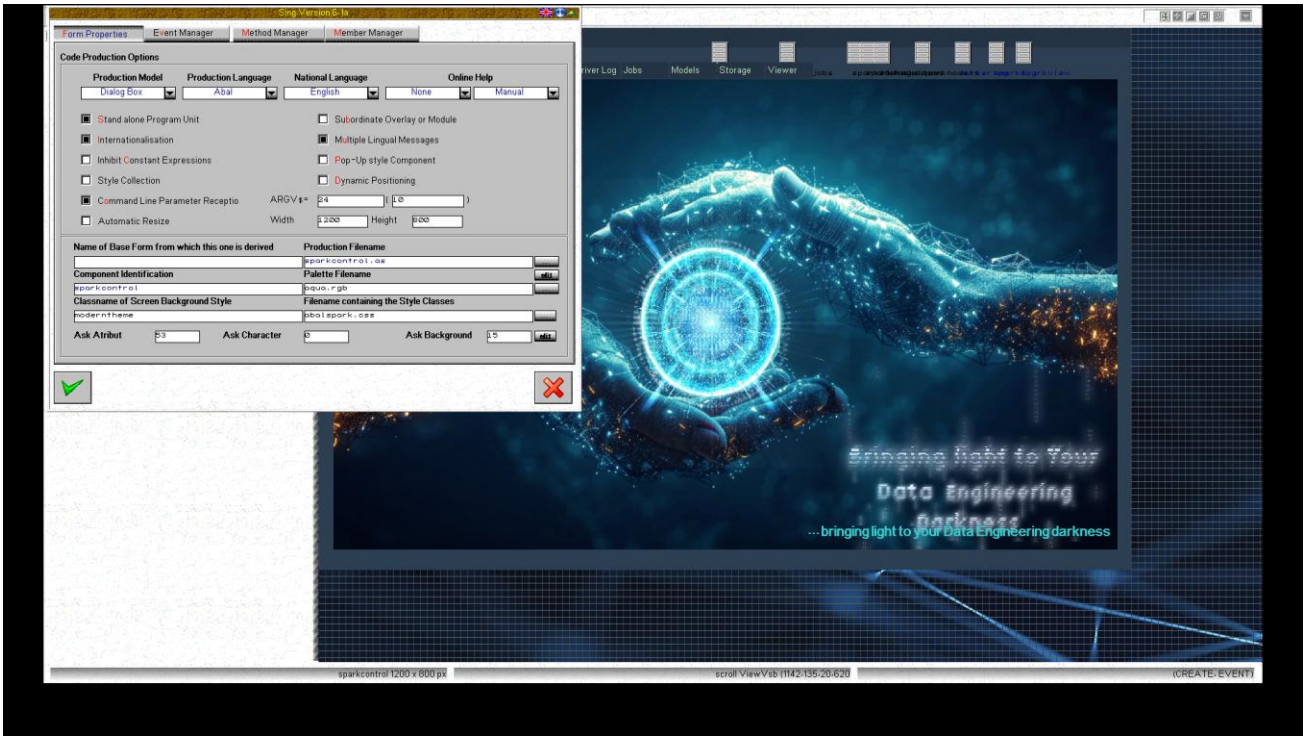
This menu option Deletes the currently selected widget.

Events

This menu option provides access to the Widget Events Menu of the currently selected widget.

FORMS PROPERTIES

The Forms Properties dialog box, shown below, will be displayed as a result of the Top Menu option Form . Properties or in response to the ENTER key event in the Forms Editor.



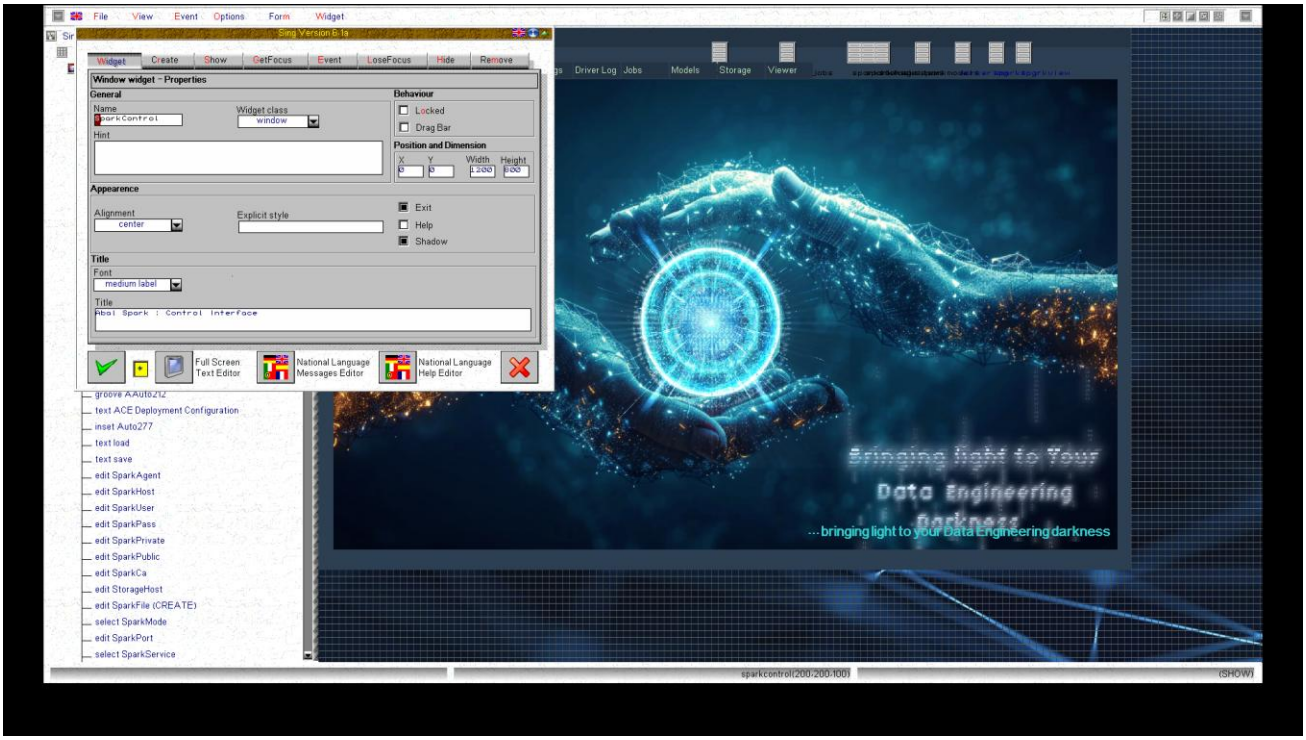
This dialogue box comprises four Tab Pages:

- Form Properties
- Event Manager
- Method Manager
- Member Manager

allowing definition of the Forms level properties, events handling configuration and the collection of public and private global methods and data members.

WIDGET PROPERTIES

The contextual Widget Properties dialog box will be displayed as a result of the Top Menu option Widget . Properties or in response to the TABULATION key event in the Forms Editor. Each widget-type-specific variation of the Wiget Properties dialog box allows consultation of the characteristics of the associated widget and access to the Text Editor for consultation and modification of the seven Widget Methods describing the widget's behaviour at the corresponding stages of the Forms Model Life Cycle : Create, Show, Get Focus, Event, Lose Focus, Hide and Remove.



Supplementary options are also available for each widget method as shown below:

Create

- **Inline without sizes** : The CREATE method extension, if defined, will be generated INLINE as required by the standard flow of Visual Forms operations. The widget position, dimensions and colours will be constant integer values.
- **Inline with sizes** : The CREATE method extension, if defined, will be generated INLINE as required by the standard flow of Visual Forms operations. The widget position, dimensions and colours will be declared as integer variable values.
- **Function without sizes** : The CREATE method extension, if defined, will be generated as a PROCEDURE that will be CALLED as required by the standard flow of Visual Forms operations. The widget position, dimensions and colours will be constant integer values.
- **Function with sizes** : The CREATE method extension, if defined, will be generated as a PROCEDURE that will be CALLED as required by the standard flow of Visual Forms operations. The widget position, dimensions and colours will be declared as integer variable values.

Show

- **Inline automatic** : The SHOW method alternative, if defined, will be generated INLINE as required by the standard flow of Visual Forms operations.
- **Inline conditional** : The SHOW method alternative, if defined, will be generated INLINE as required by explicit invocation through <widget name>.SHOW operations.

- **Function automatic** : The SHOW method alternative, if defined, will be generated as a PROCEDURE that will be CALLED as required by the standard flow of Visual Forms operations.
- **Function conditional** : The SHOW method alternative, if defined, will be generated as a PROCEDURE that will be CALLED as required by explicit invocation through <widget name>.SHOW operations.

Get Focus

- **Inline with focus** : The GET FOCUS method alternative, if defined, will be generated INLINE as required by the standard flow of Visual Forms operations.
- **Inline without focus** : The GET FOCUS method alternative, if defined, will be generated INLINE as required by explicit invocation through <widget name>.GETFOCUS operations.
- **Function with focus** : The GET FOCUS method alternative, if defined, will be generated as a PROCEDURE that will be CALLED as required by the standard flow of Visual Forms operations.
- **Function without focus** : The GET FOCUS method alternative, if defined, will be generated as a PROCEDURE that will be CALLED as required by explicit invocation through <widget name>.GETFOCUS operations.

Event

- **Inline with event** : The EVENT method extension, if defined, will be generated INLINE as required by the standard flow of Visual Forms operations.
- **Inline without event** : The EVENT method extension, if defined, will be generated INLINE as required by explicit invocation through <widget name>.EVENT operations.
- **Function with event** : The EVENT method extension, if defined, will be generated as a PROCEDURE that will be CALLED as required by the standard flow of Visual Forms operations.
- **Function without event** : The EVENT method extension, if defined, will be generated as a PROCEDURE that will be CALLED as required by explicit invocation through <widget name>.EVENT operations.

Lose Focus

- **Inline with event** : The LOSE FOCUS method alternative, if defined, will be generated INLINE as required by the standard flow of Visual Forms operations.
- **Inline without event** : The LOSE FOCUS method alternative, if defined, will be generated INLINE as required by explicit invocation through <widget name>. LOSEFOCUS operations.
- **Function with event** : The LOSE FOCUS method alternative, if defined, will be generated as a PROCEDURE that will be CALLED as required by the standard flow of Visual Forms operations.

- **Function without event** : The LOSE FOCUS method alternative, if defined, will be generated as a PROCEDURE that will be CALLED as required by explicit invocation through <widget name>. LOSEFOCUS operations.

Hide

- **Inline normal** : The HIDE method alternative, if defined, will be generated INLINE as required by the standard flow of Visual Forms operations.
- **Inline popup** : The HIDE method alternative, if defined, will be generated INLINE as required by explicit invocation through <widget name>. HIDE operations and will restore the previous screen area saved during the initial <widget name>. SHOW operation.
- **Function normal**: The HIDE method alternative, if defined, will be generated as a PROCEDURE that will be CALLED as required by the standard flow of Visual Forms operations.
- **Function popup**: The HIDE method alternative, if defined, will be generated as a PROCEDURE that will be CALLED as required by explicit invocation through <widget name>. HIDE operations and will restore the previous screen area saved during the initial <widget name>. SHOW operation.

Remove

- **Inline with event** : The REMOVE method extension, if defined, will be generated INLINE as required by the standard flow of Visual Forms operations.
- **Inline without event**: The REMOVE method extension, if defined, will be generated INLINE as required by explicit invocation through <widget name>. REMOVE operations.
- **Function with event**: The REMOVE method extension, if defined, will be generated as a PROCEDURE that will be CALLED as required by the standard flow of Visual Forms operations.
- **Function with sizes** : The REMOVE method extension, if defined, will be generated as a PROCEDURE that will be CALLED as required by explicit invocation through <widget name>. REMOVE operations.

LANGUAGE ELEMENTS

The Data Properties and Intrinsic Methods of all Widgets attached to Visual Forms models can be addressed and invoked from Widget Event Method code and from Forms Method code.

WIDGET DATA PROPERTIES

The following properties are accessible for standard visual interface widgets , where appropriate to their usage semantics.

X

This property provides access to the horizontal X position of the widget, described in pixels. When the 'with sizes' feature of the 'OnCreate' options is selected then a variable will be declared for the X position of the widget, allowing dynamic repositioning, otherwise only its constant integer value is accessible.

```
For this.X = this.X to 100 Step this.W :: this.OnShow() :: Next this.X
```

Y

This property provides access to the vertical Y position of the widget, described in pixels. When the 'with sizes' feature of the 'OnCreate' options is selected then a variable will be declared for the Y position of the widget, otherwise only its constant integer value is accessible.

```
For this.Y = this.Y to 100 step this.H :: this.OnShow() :: Next this.Y
```

W

This property provides access to the horizontal width dimension of the widget, described in pixels. When the 'with sizes' feature of the 'OnCreate' options is selected then a variable will be declared for the W value of the widget, otherwise only its constant integer value is accessible.

```
For this.W = this.W to 100 Step 10 :: this.OnShow() :: Next this.W
```

H

This property provides access to the vertical height dimension of the widget, described in pixels. When the 'with sizes' feature of the 'OnCreate' options is selected then a variable will be declared for the H value of the widget, otherwise only its constant integer value is accessible.

```
For this.H = this.H to 100 Step 10 :: this.OnShow() :: Next this.H
```

FONT

This property returns the font number associated with the widget.

FONTWIDTH

This property returns the width of the font, in pixels, associated with the widget.

FONTHEIGHT

This property returns the height of the font, in pixels, associated with the widget.

FG

This property returns the colour map index of the foreground colour associated with the widget. When the 'with sizes' feature of the 'OnCreate' options is selected then a variable will be declared for the FG colour of the widget otherwise only its constant integer value is accessible.

BG

This property returns the colour map index of the background colour associated with the widget. When the 'with sizes' feature of the 'OnCreate' options is selected then a variable will be declared for the FG colour of the widget otherwise only its constant integer value is accessible.

VALUE

This property allows R/W access to the contextual value of the following widget types:

- **Radio** : the value of the radio button set indicating the current activated button.
- **Check** : the Boolean state of the check box.
- **Select** : the numeric value of the currently selected item of the list.
- **Graph** : the current array of values.
- **Switch** : the Boolean state of the switch box.
- **Scroll** : the offset of the Scroll bar from the origin.
- **Progress** : the current value of the progression bar.
- **Form** : the complete buffer of data to be tabulated.
- **Edit** : the string buffer of the edit box.

Other widget types will raise a syntax error.

BUFFER

This property allows R/W access to the contextual string **buffer** associated with the widget type.

LIMIT

This property allows R/W access to the integer **limit** value of Scroll Bar widget types, representing the size of the bar compared to the total quantity.

MAX

This property allows R/W access to the integer **max** value of Scroll Bar widget types, and of Progress Bar widget types.

LINE

This property allows positioning of the current line of Form Widgets.

LINES

This property returns the number of lines of the associated Form Widget.

```
For x = 1 to this.Lines :: this.Line(x) :: Next x
```

SIZE

This property returns the calculated usage size of Edit Box Widgets or the Y limit of others.

COLUMNS

This property returns the number of columns of the associated Form Widget.

```
For x = 1 to this.Columns
Select x
Case 1 :: this.column(1) = “value”
EndSel
Next x
```

COLUMN

This property allows R/W access to the string value of the indicated column of the currently select Line of the associated Form Widget.

```
For x = 1 to this.Columns
Select x
Case 1 :: this.column(1) = “value”
EndSel
Next x
```

COLUMNSIZE

This property returns the width/size of the string value buffer of the indicated of column of the associated Form Widget.

POSITION

This property returns the calculated X position of the start of the indicated column of the associated Form Widget.

PAYLOAD

This property allows access to the current national language payload field of the following widgets:

- **Text** : returns the constant or variable value of the Text Label.
- **Window** : returns the constant or variable value of the Window Title.
- **Check** : returns the constant or variable value of the Check Label.
- **Radio** : returns the constant or variable value of the Radio Label.
- **Switch** : returns the constant or variable value of the Switch Message.
- **Other :** returns the string value of the Payload field.

FORMAT

This property returns the contextual OPEN ABAL PRINT/ASK format string associated with the Widget value of the following widget types:

- **Data** : returns the widget properties specified format field.

- **Form**: returns the widget properties specified format field.
- **Edit** : returns the widget properties specified format field or (E)
- **Other** : returns the widget properties specified format field or (E)

HOTKEY

This property returns the HOTKEY Trigger value for the associated Widget.

FOCUS

This property returns the FOCUS ORDER number of the associated Widget.

PAGE

This property returns the TAB PAGE number of the associated Widget.

STYLE

This property returns the string value of the **stylesheet** field of the associated Widget.

HINT

Returns the string value of the current national language indexed hint value of the associated Widget.

FORMSORT

This property returns the value of the **Form Sort** field of the associated Widget.

TABSORT

This property returns the value of the **Tab Page Sort** field of the associated Widget.

LINESORT

This property returns the value of the **Line Sort** field of the associated Widget.

STARTSORT

This property returns the value of the **Start Sort** field of the associated Widget.

SORT

This property returns the value of the **Form Sort** field of the associated Widget.

OFFSET

This property returns the offset of the TAB BUTTON from the left edge of the TAB PAGE widget.

ACCESS

This property returns the value of the **Access** field of the associated Widget

HOTKEYS

This property returns the value of the **Hotkeys** field of the associated Widget

LABELS

This property returns the string value of the **Labels** field of the associated Form Widget

FORMATS

This property returns the string value of the **Format** field of the associated Form Widget

WIDGET METHODS

The following methods may be invoked for standard visual interface widgets , where appropriate to their usage semantics.

CREATE

This widget method launches invocation of the contextual, Form Specific, construction assistant method of the associated Widget.

SHOW

This widget method launches invocation of the contextual, Form Specific, display assistant method of the associated Widget.

GETFOCUS

This widget method launches invocation of the contextual, Form Specific, focus management assistant method of the associated Widget.

EVENT

This widget method launches invocation of the contextual, Form Specific, event management assistant method of the associated Widget.

LOSEFOCUS

This widget method launches invocation of the contextual, Form Specific, focus relinquishment assistant method of the associated Widget.

HIDE

This widget method launches invocation of the contextual, Form Specific, disappearance assistant method of the associated Widget.

REMOVE

This widget method launches invocation of the contextual, Form Specific, destruction assistant method of the associated Widget.

ONCREATE

This widget method launches invocation of the Widget Type Specific construction method of the associated Widget.

ONSHOW

This widget method launches invocation of the Widget Type Specific display method of the associated Widget.

ONFOCUS

This widget method launches invocation of the Widget Type Specific focus management method of the associated Widget.

ONEVENT

This widget method launches invocation of the Widget Type Specific event management method of the associated Widget.

ONHIDE

This widget method launches invocation of the Widget Type Specific disappearance method of the associated Widget.

ONREMOVE

This widget method launches invocation of the Widget Type Specific destruction method of the associated Widget.

ONSET

This widget method launches invocation of the remote Set method of the associated Data Module or Overlay Widget

ONGET

This widget method launches invocation of the remote Get method of the associated Data Module or Overlay Widget

DROP

This widget method launches invocation of Pop-Up memory release method of the associated Widget

ALLOCATE

This widget method launches invocation of Pop-Up memory allocation method of the associated Widget

TRIGGER

The invocation of this method selects the associated Tab Page Widget as the active Tab Page.

```
Widget.Trigger()
```

FREEZE

This widget method signals the start of graphical output freeze for the associated Widget.

THAW

This widget method signals the end of graphical output freeze for the associated Widget.

FILL

This widget method fills the visual screen/output zone occupied by the associated with the defined foreground colour value of the Widget.

WIPE

This widget method fills the visual screen/output zone occupied by the associated with the defined background colour value of the Widget.

RESET

This widget method resets the coordinates and size of a dynamic dimensional Widget to its original position and dimensions.

PUT

This widget method transmits the contents of an allocated Pop-Up buffer, of the associated Widget, to the output or screen

ACTION

This widget method launches invocation of the Action method of the associated Wondow Widget.

VIEWPORT

This widget method launches creation of a Visual View Port for the position, dimensions and font characteristics of the associated Widget.

PRESS

Invocation of this widget method for Button Widgets displays the button in depressed mode.

HELP

The invocation of this method displays the help bubble of the associated Widget.

```
Widget.Help()
```

ENABLE

The invocation of this method renders the associated Widget enabled.

```
Widget.Enable()
```

DISABLE

The invocation of this method renders the associated Widget disabled.

```
Widget.Disable()
```

INHIBIT

The invocation of this method renders the associated Widget inhibited.

```
Widget.Inhibit()
```

FILE WIDGET DATA PROPERTIES

The following properties are accessible for file and database widgets , where appropriate to their usage semantics.

INDEX

This property allows R/W access to the string buffer representing the PRIMARY KEY of a sequential indexed or multiple index database table Widget.

ERROR

The ERROR property stores the result of all FILE WIDGET method invocations for the corresponding File Widget and may be inspected to determine the success or failure of the most recent operation.

HANDLE

The HANDLE property allows R/W access to the file handle that has been allocated, via EVENT 77, for use by the ASSIGN method to identify operations on this File Widget.

QUESTION

This property is used to store the QUERY for POSIT, COUNT, COLLECT, EXEC and SELECT File Methods.

RESPONSE

This property returns the number of result elements returned by POSIT, COUNT, COLLECT, EXEC and SELECT File Method queries.

LENGTH

This property manages the returns useful length of data exchange buffers for SEARCH, FIRST, UP,PREVIOUS, DOWN,NEXT,LAST instructions.

MARKER

This property returns the value of the legacy index marker for SEARCH, FIRST, UP,PREVIOUS, DOWN,NEXT,LAST instructions.

FLAG

This property allows R/W access to the sector value of a direct access data widget.

DIRECTION

This property allows R/W access to the direction value of a direct access data widget.

SECTOR

This property allows R/W access to the sector value of a direct access data widget.

FILE WIDGET METHODS

The following methods may be invoked for standard visual interface widgets , where appropriate to their usage semantics.

The methods available for multiple indexed file tables have been extended to include the new SQL instructions available with OPENABAL and INXSQL.

ASSIGN

The ASSIGN method will be implicitly invoked for all FILE WIDGETS from the ONCREATE method of the parent Visual Form.

This widget file method prepares the file widget for subsequent file-oriented methods. In most cases a traditional OPENABAL ASSIGN instruction will be generated for a file handle that has been allocated using the EVENT 77 mechanism.

This instruction will now detect and make use of an explicit braced assign handle number, appended to the assigned filename property, for DATABASE and SEQUENTIAL/MULTIPLE INDEXED file widgets only. This is essential to allow the relationship between DATABASES and their TABLES.

The filename property is defined via the Widget Properties dialog box of the Form Editor.

In the case of a database File Widget the filename syntax should be:

```
“username:password@hostname:port:basename(number)”
```

The constant integer value of the extracted number will be used to ASSIGN the database:

```
handle = number

ASSIGN=handle,” username:password@hostname:port:basename”,DB,WR:next,error
```

In the case of a sequential or multiple indexed File Widget to be declared as subordinate to an explicit database File Widget, the filename syntax would be:

```
“tablename(number)”
```

and to connect the indexed or multiple indexed ASSIGN to the database.

```
handle = event(77)

ASSIGN=handle,” tablename”,MC(number),WR:next,error,columntable
```

OPEN

The OPEN method will perform a contextual OPEN instruction of the appropriate type required by the nature of the File Widget.

CLOSE

The CLOSE method will perform a contextual CLOSE instruction of the appropriate type required by the nature of the File Widget.

CFILE

The **CFILE** method will perform a contextual **CFILE** instruction of the appropriate type required by the nature of the File Widget.

DFILE

The **DFILE** method will perform a contextual **DFILE** instruction of the appropriate type required by the nature of the File Widget.

RENAME

The RENAME method will perform a contextual **RENAME** instruction of the appropriate type required by the nature of the File Widget.

LOCK

The **LOCK** method will perform a standard **SEARCH.M** instruction for the corresponding database table Widget.

ATB

The ATB method will perform a contextual ATB instruction of the appropriate type required by the nature of the File Widget.

UPDATE

The UPDATE method will perform an INSERT / MODIF record update construction for sequential indexed and multiple indexed File Widgets.

INSERT

The INSERT method will perform the standard OPENABAL INSERT instruction for sequential indexed and multiple indexed File Widgets.

The data record of the corresponding File Widget should contain the record field values.

MODIF

The MODIF method will perform the standard OPENABAL MODIF instruction for sequential indexed and multiple indexed File Widgets.

The data record of the corresponding File Widget should contain the record field values.

DELETE

The DELETE method will perform the standard OPENABAL DELETE instruction for sequential indexed and multiple indexed File Widgets.

SEARCH

The SEARCH method will perform the standard OPENABAL SEARCH instruction for sequential indexed and multiple indexed File Widgets.

The optional .M, .L and .ML extensions to this File Widget method provide for the traditional record locking and variable record length handling.

The data record of the corresponding File Widget will contain the record fields.

FIRST

The FIRST method will perform the OPENABAL INXS FIRST instruction for sequential indexed and multiple indexed File Widgets.

The optional .M, .L and .ML extensions to this File Widget method provide for the traditional record locking and variable record length handling.

The data record of the corresponding File Widget will contain the record fields.

UP

The UP method will perform the standard OPENABAL UP instruction for sequential indexed and multiple indexed File Widgets.

The optional .M, .L and .ML extensions to this File Widget method provide for the traditional record locking and variable record length handling.

The data record of the corresponding File Widget will contain the record fields.

PREVIOUS

The PREVIOUS method will perform the standard OPENABAL UP instruction for sequential indexed and multiple indexed File Widgets.

The optional .M, .L and .ML extensions to this File Widget method provide for the traditional record locking and variable record length handling.

The data record of the corresponding File Widget will contain the record fields.

DOWN

The DOWN method will perform the standard OPENABAL DOWN instruction for sequential indexed and multiple indexed File Widgets.

The optional .M, .L and .ML extensions to this File Widget method provide for the traditional record locking and variable record length handling.

The data record of the corresponding File Widget will contain the record fields.

NEXT

The NEXT method will perform the standard OPENABAL DOWN instruction for sequential indexed and multiple indexed File Widgets.

The optional .M, .L and .ML extensions to this File Widget method provide for the traditional record locking and variable record length handling.

The data record of the corresponding File Widget will contain the record fields.

LAST

The LAST method will perform the OPENABAL INXS LAST instruction for sequential indexed and multiple indexed File Widgets.

The optional .M, .L and .ML extensions to this File Widget method provide for the traditional record locking and variable record length handling.

The data record of the corresponding File Widget will contain the record fields.

KEY

The KEY method allows declaration of SQL table column characteristics using the standard OPENABAL KEY instruction for sequential indexed and multiple indexed File Widgets.

CKEY

The CKEY method allows creation of SQL table columns using the standard OPENABAL CKEY instruction for sequential indexed and multiple indexed File Widgets.

FKEY

The FKEY method allows redefinition of SQL table columns using the standard OPENABAL FKEY instruction for sequential indexed and multiple indexed File Widgets.

Since OPENABAL the PRIMARY INDEX column may also be redefined in this way.

RKEY

The RKEY method allows reset of SQL table column descriptions using the standard OPENABAL RKEY instruction for sequential indexed and multiple indexed File Widgets.

NKEY

The NKEY method allows renaming of SQL table column descriptions using the standard OPENABAL NKEY instruction for sequential indexed and multiple indexed File Widgets.

LKEY

The LKEY method allows the of SQL table columns of multiple indexed and database File Widgets to be listed using the standard OPENABAL LKEY instruction.

This instruction expects the traditional (F) and (N) options indicating first and next element retrieval.

LINK

The LINK method allows declaration of SQL table indexes using the standard OPENABAL LINK instruction for multiple indexed File Widgets.

CLINK

The CLINK method allows creation or deletion of SQL table indexes using the standard OPENABAL CLINK instruction for multiple indexed File Widgets.

LLINK

The LLINK method allows the of SQL table indexes of multiple indexed and database File Widgets to be listed using the standard OPENABAL LLINK instruction.

This instruction expects the traditional (F) and (N) options indicating first and next element retrieval.

COUNT

The COUNT instruction allows counting of matching SQL Table records of multiple indexed and database File Widgets using the standard OPENABAL COUNT instruction.

The query may be provided as a parameter to this Method or will be taken from the QUESTION property of the corresponding File Widget.

The Method will return the number of records selected in the RESPONSE property of the corresponding File Widget.

POSIT

The POSIT instruction allows selection of a collection of SQL Table records of multiple indexed and database File Widgets using the standard OPENABAL POSIT instruction.

This instruction allows the standard .D and OPENABAL extensions of .C and .S variations.

The query may be provided as a parameter to this Method or will be taken from the QUESTION property of the corresponding File Widget.

The Method will return the number of records selected in the RESPONSE property of the corresponding File Widget.

DISTINCT

The POSIT instruction allows selection of a distinct of collection of SQL Table records of multiple indexed and database File Widgets using the standard OPENABAL POSIT instruction.

This instruction allows the standard .D and OPENABAL extensions of .C and .S variations.

The query may be provided as a parameter to this Method or will be taken from the QUESTION property of the corresponding File Widget.

The Method will return the number of records selected in the RESPONSE property of the corresponding File Widget.

COLLECT

The COLLECT instruction allows selection and extraction of the primary index values of records matching the request query of the collection of SQL Table records of multiple indexed and database File Widgets using the standard OPENABAL COLLECT instruction.

The query may be provided as a parameter to this Method or will be taken from the QUESTION property of the corresponding File Widget.

The Method will return the number of records selected in the RESPONSE property of the corresponding File Widget.

The method will allocate the response array containing the retrieve collection of primary index values that are to be accessed using the LOCATE File Widget Method.

LOCATE

The LOCATE instruction allows retrieval of the results of a COLLECT instruction using the index value passed as parameter to locate the indexed element in the result set and the standard OPENABAL SEARCH instruction.

The data record of the corresponding File Widget will contain the record fields.

EXEC

The EXEC instruction submits a standard and complete SQL query to multiple indexed and database File Widgets using the OPENABAL INXS EXEC instruction.

The query may be provided as a parameter to this Method or will be taken from the QUESTION property of the corresponding File Widget.

No response, other that operational error status will be returned by this method.

SELECT

The SELECT instruction submits a standard and complete SQL query to multiple indexed and database File Widgets using the OPENABAL INXS SELECT instruction.

The query may be provided as a parameter to this Method or will be taken from the QUESTION property of the corresponding File Widget.

The number of records in the result set will be returned in the RESPONSE property.

The result set will be allocated and attached to the result set pointer, the individual elements of which are to be retrieved using the CONNECT and RETRIEVE Methodes of the corresponding File Widget.

```
Dcl mybuffer$=256

Field=m,mybuffer

Dcl myfields$=64(4)

Field=m

 MyTable.Connect(mybuffer)

 MyTable.Select("select name,town,telephone,email from table where name > ' '")

 For I = 1 to MyTable.Records

    MyTable.Retrieve(l)

 Next l
```

RETRIEVE

The RETRIEVE method allows retrieval of the results of a SELECT instruction of multiple indexed and database File Widgets using the index value passed as parameter to locate the indexed element in the result set.

The data record of to receive the output data must have been prepared using the CONNECT method for the corresponding File Widget.

CONNECT

The CONNECT method allows preparation of the output fields buffer for retrieval of the results of a SELECT instruction of multiple indexed and database File Widgets.

RECORD

The RECORD method allows declaration of an SQL table output record using the standard OPENABAL RECORD instruction for database File Widgets.

LFILE

The LFILE method allows SQL tables of database File Widgets to be listed using the standard OPENABAL LFILE instruction.

This instruction expects the traditional (F) and (N) options indicating first and next element retrieval.

JOIN

The JOIN method allows declaration of an inter table SQL table JOIN using the standard OPENABAL JOIN instruction for database File Widgets.

CJOIN

The CJOIN method allows creation of an inter table SQL table JOIN using the standard OPENABALC JOIN instruction for database File Widgets.

DJOIN

The DJOIN method allows deletion of an inter table SQL table JOIN using the standard OPENABAL DJOIN instruction for database File Widgets.

RJOIN

The RJOIN method allows renaming of an inter table SQL table JOIN using the standard OPENABAL RJOIN instruction for database File Widgets.

LJOIN

The LJOIN method allows the inter table SQL table JOINs of database File Widgets to be listed using the standard OPENABAL LJOIN instruction.

This instruction expects the traditional (F) and (N) options indicating first and next element retrieval.

ONCREATE

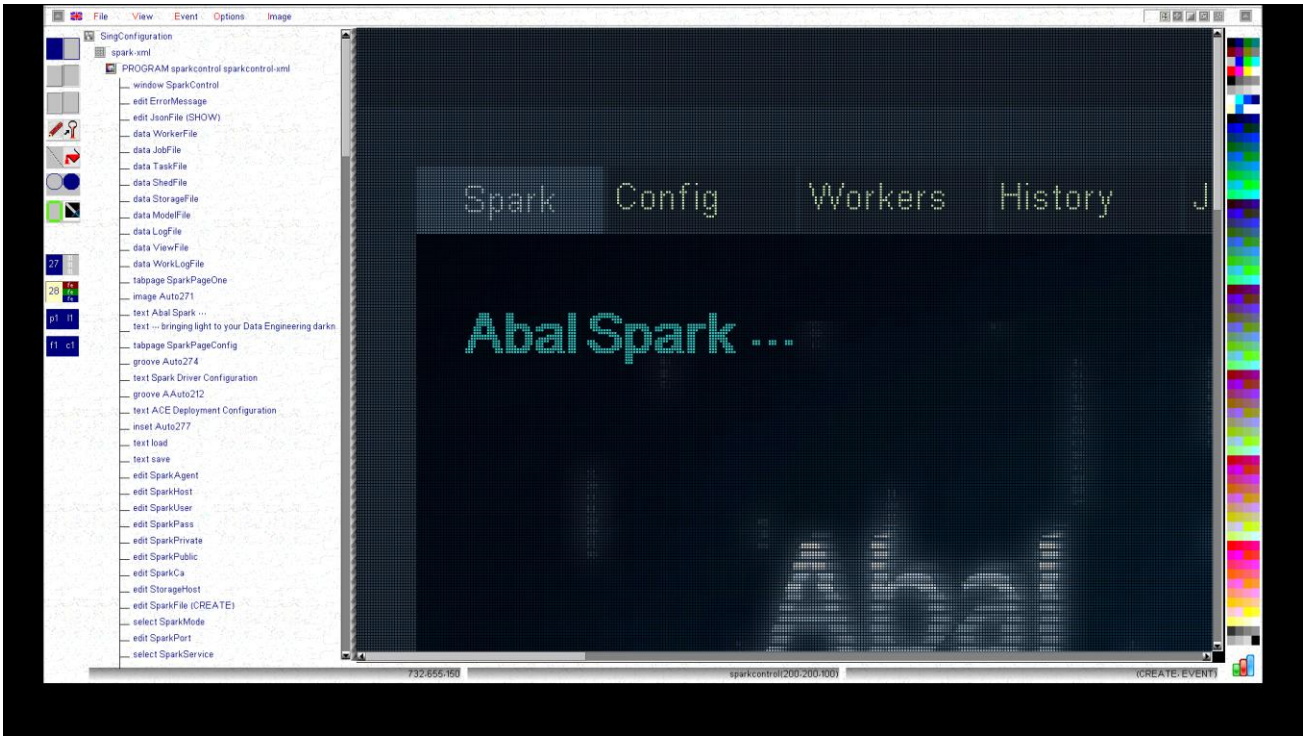
Invocation of the **ONCREATE** method performs the standard construction required for the specific Data Widget type.

ONREMOVE

Invocation of the **ONREMOVE** method performs the standard destruction required for the specific Data Widget type.

IMAGE EDITOR

The Image Editor of the Open Abal Graphics Design Tool, SING, provides a collection of tools allowing inspection and modification of standard PNG, BMP, GIF and JPEG image files, at the colour pixel level.



NAVIGATION

The Pixel Viewer of the Image Editor may be navigated using the keyboard as follows:

- DOWN ARROW (CTRL E) scrolls the image down by one row of pixels.
- UP ARROW (CTRL K) scrolls the image up by one row of pixels.
- LEFT ARROW (CTRL H) scrolls the image left by one column of pixels.
- RIGHT ARROW (CTRL F scrolls the image right by one column of pixels.
- PAGE DOWN (CTRL C) scrolls the image by half a page downwards.
- PAGE UP (CTRL R) scrolls the image by half a page upwards.
- HOME (CTRL L) scrolls the image up to the top left corner.
- END (CTRL B) scrolls the image down to the bottom right corner.
- TABULATION (CTRL I) scrolls the image by half a page to the right.

The Pixel Viewer of the Image Editor may be navigated using the MOUSE as follows:

- LEFT BUTTON DRAG, to the left or right, will scroll the image left or right accordingly.
- LEFT BUTTON DRAG, upwards or downwards, will scroll the image accordingly.

TOOLS

The contextual Tool Bar offers the following PIXEL modification and interaction tools.

Fill Rectangle

This tool will fill the rectangle starting from the pixel selected by the mouse button down and ending at the pixel selected by the mouse button up. The LEFT button of the MOUSE will select the foreground colour for the fill and the RIGHT button the background colour.

Rectangle Frame

This tool will draw a rectangular frame starting from the pixel selected by the mouse button down and ending at the pixel selected by the mouse button up. The LEFT button of the MOUSE will select the foreground colour for the fill and the RIGHT button the background colour.

Inset Frame

This tool will draw an inset rectangular frame starting from the pixel selected by the mouse button down and ending at the pixel selected by the mouse button up. The standard GUI colours will be used.

Outset Frame

This tool will draw an outset rectangular frame starting from the pixel selected by the mouse button down and ending at the pixel selected by the mouse button up. The standard GUI colours will be used.

Ridge Frame

This tool will draw a rectangular ridge frame starting from the pixel selected by the mouse button down and ending at the pixel selected by the mouse button up. The standard GUI colours will be used.

Groove Frame

This tool will draw a rectangular groove frame starting from the pixel selected by the mouse button down and ending at the pixel selected by the mouse button up. The standard GUI colours will be used.

Pencil

This tool will draw a single pixel at the coordinates of the mouse button CLICK.

The LEFT button of the MOUSE will select the foreground colour for the fill and the RIGHT button the background colour.

Eye Dropper

This tool will copy a single pixel colour at the coordinates of the mouse button CLICK.

The LEFT button of the MOUSE will copy to the foreground colour and the RIGHT button to the background colour.

Circular Frame

This tool will draw a circular frame starting from the pixel selected by the mouse button down and ending at the pixel selected by the mouse button up. The LEFT button of the MOUSE will select the foreground colour for the fill and the RIGHT button the background colour.

Disk Fill

This tool will fill the circular disk starting from the pixel selected by the mouse button down and ending at the pixel selected by the mouse button up. The LEFT button of the MOUSE will select the foreground colour for the fill and the RIGHT button the background colour.

Pixel Selection

This tool allows a pixel zone to be defined for the Image Menu Copy Pixels function.

Pixel Pointer

This tool allows a pixel coordinate to be defined for the Image Menu Paste Pixels function.

MENUS

Image Menu

This Top Menu option provides access to all Image specific menu options.

Copy

This menu option copies the currently selected pixels **to** the Image Editor clipboard.

Paste

This menu option pastes the currently selected pixels **from** the Image Editor clipboard to the current pixel Cursor position.

Rotate 90°

This menu option rotates the image clockwise by 90°.

Rotate 180°

This menu option rotates the image clockwise by 180°

Rotate 270°

This menu option rotates the image clockwise by 270°

Change Colour

This menu option provides access to the Colour Transformation dialog box allowing transformation of collections of pixels of the same colour.

Grey Scale

This menu option transforms the image to its grey scale.

Vertical Mirror

This menu option reflects the image along the vertical axis.

Horizontal Mirror

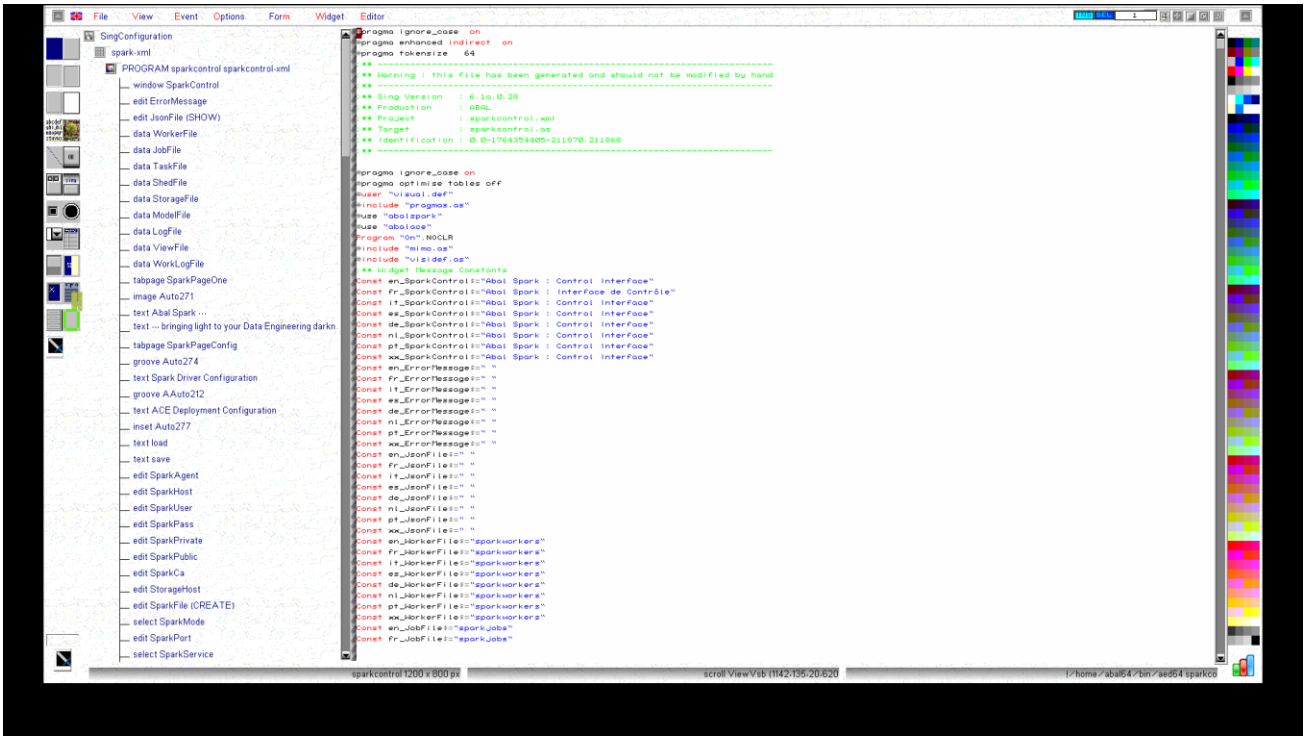
This menu option reflects the image along the horizontal axis.

Dimensions

This menu option provides access to the Image Dimensions dialog box allowing the Image dimensions to be changed or distorted.

TEXT EDITOR

The internal text editor is based on a mouse enable version of the standard ABAL text editor, AED with syntactical coloration.



NAVIGATION

The document in the Text Editor may be navigated using the keyboard as follows:

- DOWN ARROW (CTRL E) moves the text cursor down one line.
- UP ARROW (CTRL K) moves the text cursor up one line.
- LEFT ARROW (CTRL H) moves the text cursor left by one column.
- RIGHT ARROW (CTRL F) moves the text cursor right by one column.
- PAGE DOWN (CTRL C) scrolls the text cursor by half a page downwards.
- PAGE UP (CTRL R) scrolls the text cursor by half a page upwards.
- END (CTRL B) toggles between the start and end of the current line.
- TABULATION (CTRL I) moves the text curser right to the next tabulation position.
- ENTER (CTRL M) inserts a newline at the current text cursor position.
- CTRL Z attempts to load the filename under the current text cursor position.
- CTRL O toggles between insert and overwrite data entry mode.
- CTRL K deletes from the current text cursor position to the end of the line.
- DELETE deletes the character at the text cursor position.
- ESCAPE presents the command window allowing standard AED commands to be formulated.

The document in the Text Editor may be navigated using the MOUSE as follows:

- LEFT BUTTON DRAG, upwards or downwards, will scroll the image accordingly.
- LEFT BUTTON CLICK will position the text cursor accordingly.

MENUS

Editor Menu

This Top Menu option provides access to all Text Editor specific menu options.

Find

This menu option provides access to the Locate Text dialog box.

Replace

This menu option provides access to the Replace Text dialog box.

Syntax

This menu option provides access to the Widget Syntax Composer dialog box.

Resolve

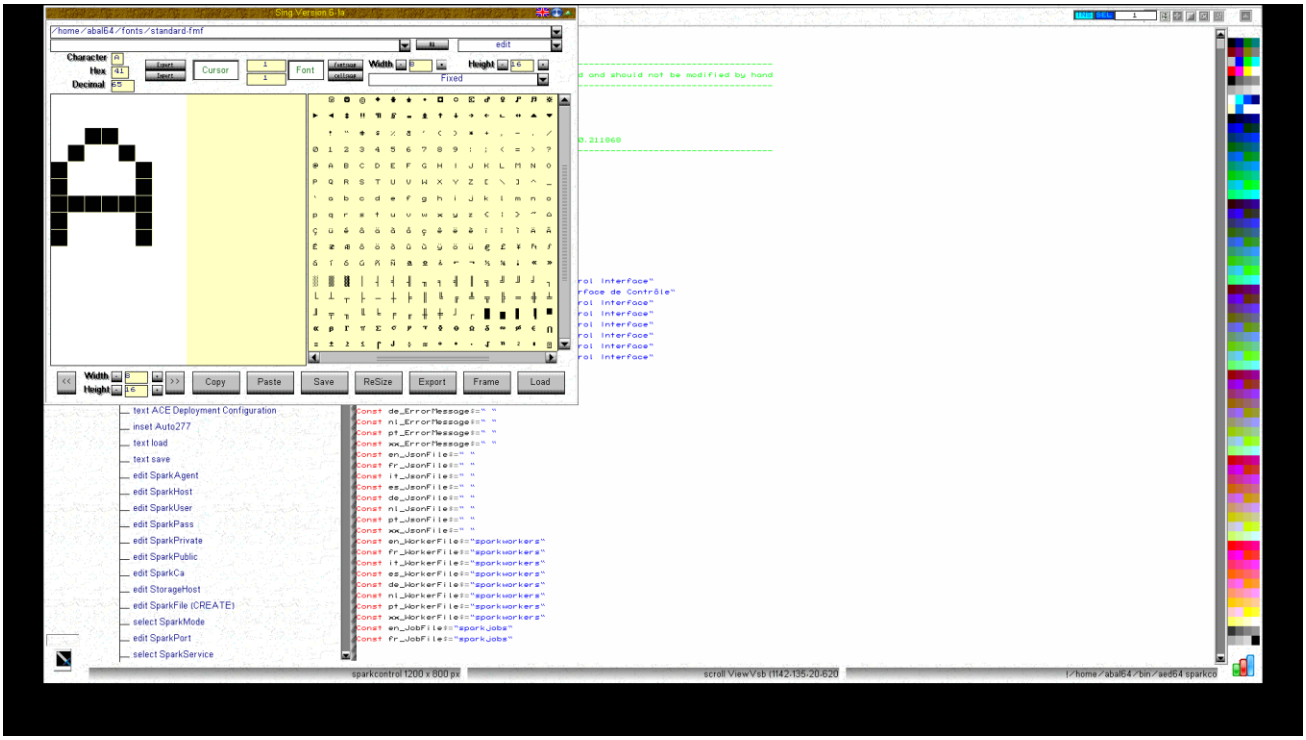
This menu option resolves the token under the Text Editor Cursor as a symbolic widget name and opens the corresponding Widget Properties dialog box.

Hyperlink

This menu option resolves the token under the Text Editor Cursor as a filename and loads file if it exists and is accessible.

FONT EDITOR

The Font Editor dialog box may be activated by the “Font” option of the “View” menu and allows consultation and modification of the standard FMF 256 entry graphical text font files.



The Font Editor comprises the following zones:

- The Character Matrix
- The ASCII Table
- The upper tool section.
- The lower tool and function section.

Text fonts may be of fixed dimensions, where each character matrix has the same width and height. Proportional spaced fonts may also be handled where the character matrix varies according to each character’s individual display requirements.

NAVIGATION and FUNCTIONS

The drop-down font file name selector allows a loaded font to be selected for inspection or modification using the Font Editor.

Text Characters may be loaded into the Character Matrix for consultation and modification by clicking on the corresponding character of the ASCII Table with the LEFT button of the MOUSE.

When the Character Matrix cursor has been activated, using the corresponding tool of the upper tool section, it may be moved using the standard cursor keys, LEFT, RIGHT, UP and DOWN.

The space bar allows the state of a pixel in the Character Matrix to be toggle between active and inactive.

A LEFT button MOUSE click on a pixel of the Character Matrix will also toggle the pixel state.

The Global font dimension controls in the upper tool section may be used in conjunction with the RESIZE function button to resize all characters in the font.

The character font dimension controls in the left-hand corner of the lower tool section may be used to re-dimension the currently loaded character.

Character Matrix data may be copied and pasted between different character positions of the ASCII table using the corresponding COPY and PASTE functions of the lower tool section.

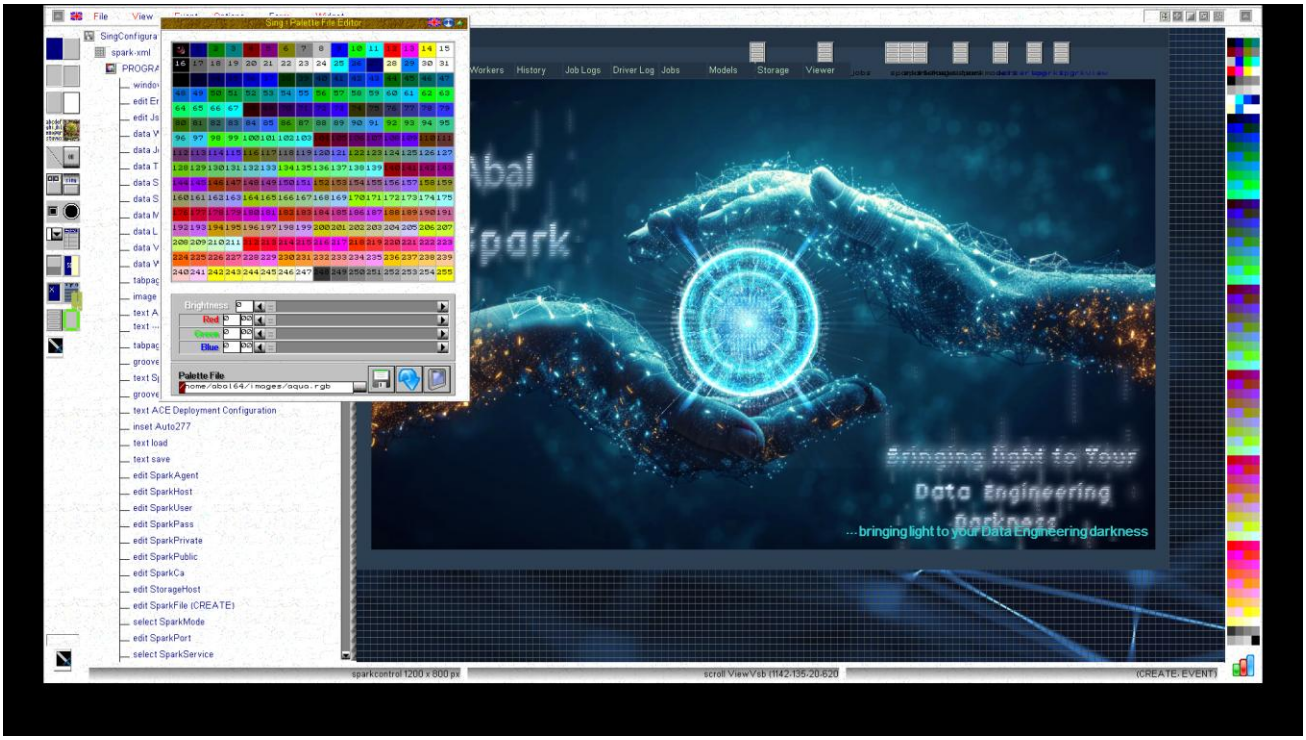
The entire collection of Character definitions may be loaded and saved using the LOAD and SAVE functions of the lower tool section.

The EXPORT function of the lower tool section allows production of an IMAGE file rendering of the current state of the font's characters.

The FRAME function of the lower tool section will generate the full collection of single- and double-line drawing symbols in the standard positions.

PALETTE EDITOR

The Palette button at the bottom of the Colour Palette bar provides access to the Palette File Editor dialog box allowing the currently loaded palette file to be accorded to your applications precise colour needs.



FUNCTIONS

Palette

The 256 Colour Palette entries are displayed as a 16 by 16 array of colour squares which may be selected for modification by a simple LEFT button MOUSE click on the required colour entry.

Save

Saves the current palette data to the selected palette filename.

Load

Loads the palette data from the selected palette filename.

Refresh Screen

Redraws the Screen with the current palette state.

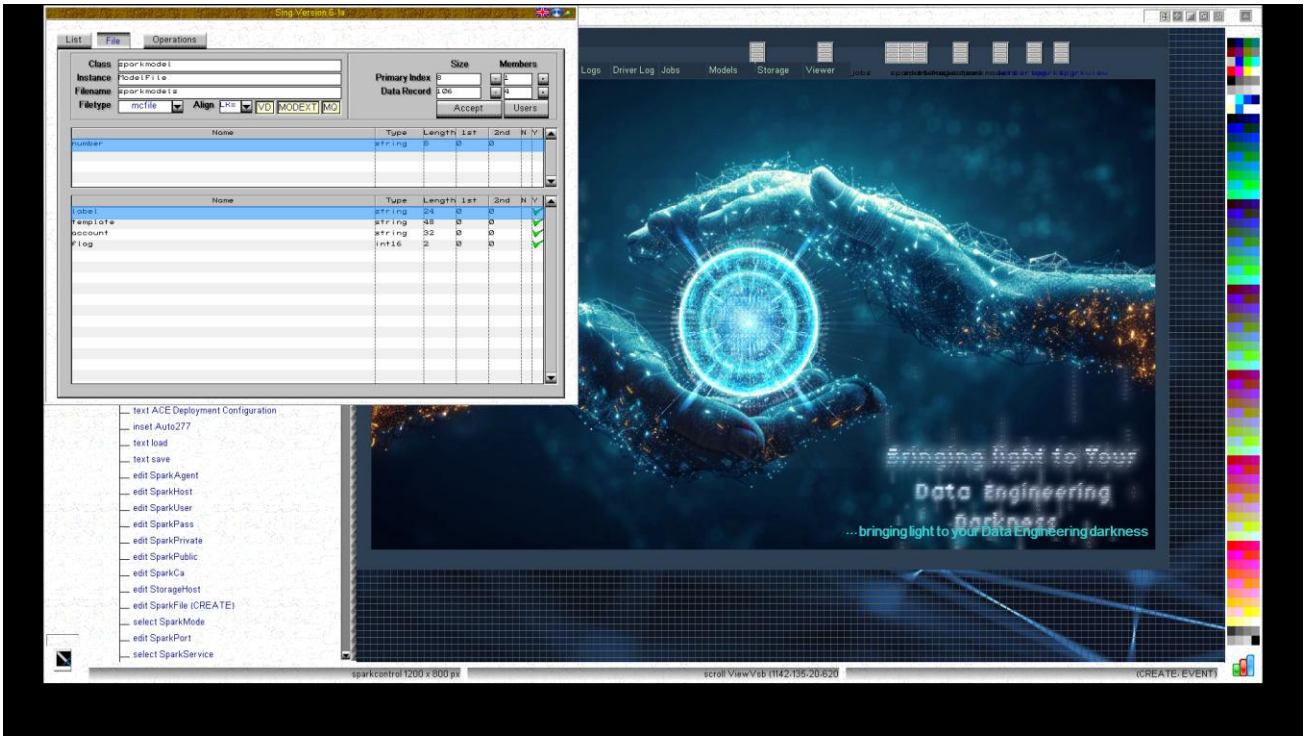
Adjust Colour

The plus and minus buttons and the scroll bar of the brightness and colour component controls allow each individual colour value to be adjusted.

DATABASE EDITOR

Overview

The database editor dialog box is launched by the Top Menu option View . Database and offers three Tab Pages for the management of database table definitions that can be used in the construction of the Open Abal Visual Applications.



List

This Tab Page presents the scrollable list of Database Tables currently defined and managed by the Database Editor.

A Database Table can be selected to be the current Table by a simple LEFT button MOUSE click on a table's name in the list.

The Copy File button will copy the description of the currently selected table to the Database Editor clipboard and will display the Paste File button.

The Paste File button allows the contents of the Database Editor clipboard to be pasted into a selected empty slot in the list.

The Delete File button allows the currently selected Table definition to be deleted and removed from the Database Editor's list.

File

This Tab Page allows consultation and modification of the currently selected Table definition in terms of its basic table characteristics and its Primary Key and Data Record Members.

Class

The string value of this field is the unique class name of this Database Table.

Instance

The string value of this field is the default instance name of this Database Table.

Filename

The string value of this field is the default filename or database table name of this Database Table.

Filetype

The file type can be one of the following proposed values:

- Variable : the data entity is a simple variable
- Instance : the data entity is a structured instance
- Visual : the data entity is a visual instance
- Sifile : the data entity is a sequential indexed file
- Mcfile : the data entity is a multiple indexed file
- Vmemory : the data entity is a virtual memory file
- Sqfile : the data entity is a sequential file
- Dbfile : the data entity is a master database file
- Adfile : the data entity is a sectorial direct access file.
- Visualgroup : the data entity is a visual group.
- Occi : the data entity is an OCCI endpoint.
- Mirror : the data entity is a mirrored OCCI endpoint.
- Replicate : the data endpoint is a replicated OCCI endpoint.

Align

The value of this field applies to the sequential and multiple indexed data entities only and defines the alignment of the primary index.

VD

This switch allows variable length data record to be selected.

MODEXT

This switch allows use of MODEXT to be selected (Deprecated).

MQ

This switch allows use of legacy marker index management to be selected (Deprecated).

Primary Index

This field describes the size and member count of the primary index of sequential and multiple indexed database tables.

Data Record

This field describes the size and member count of the data record of all data entities.

Primary Index Members

The first table of members shows the details of the data members comprising the primary index (if applicable).

Data Record Members

The second table of members shows the details of the data members comprising the data record.

In both the Primary Index Member List and the Data Record Member List the corresponding member count field will control the number of valid entries.

A member may be altered by selecting the required name column and typing in a traditional data definition (without the DCL keyword), then by pressing TABULATION the input string will be processed to ventilate the corresponding columns values.

The “N” and “Y” column entries may be used to toggle:

The definition of the data record member as a secondary index.

The inclusion of the data member in operations performed on the final Tab Page.

Operations

This Tab Page allows a collection of self-explanatory operations to be performed on the currently selected Data Entity.

- Generate File Mask : generates a Visual Form model allowing data input and record management operations for the data members selected by the “N” column of the preceding member list tables.
- Generate File List : generates a Visual Form model presenting a consultation and input list of the data members selected by the “N” column of the preceding member list tables.
- Generate File Printout : generates a Visual Page model allowing print out of the list of the data members selected by the “N” column of the preceding member list tables.
- Import : imports data entity descriptions using the preceding parameters.
- Export : exports data entity descriptions using the preceding parameters.

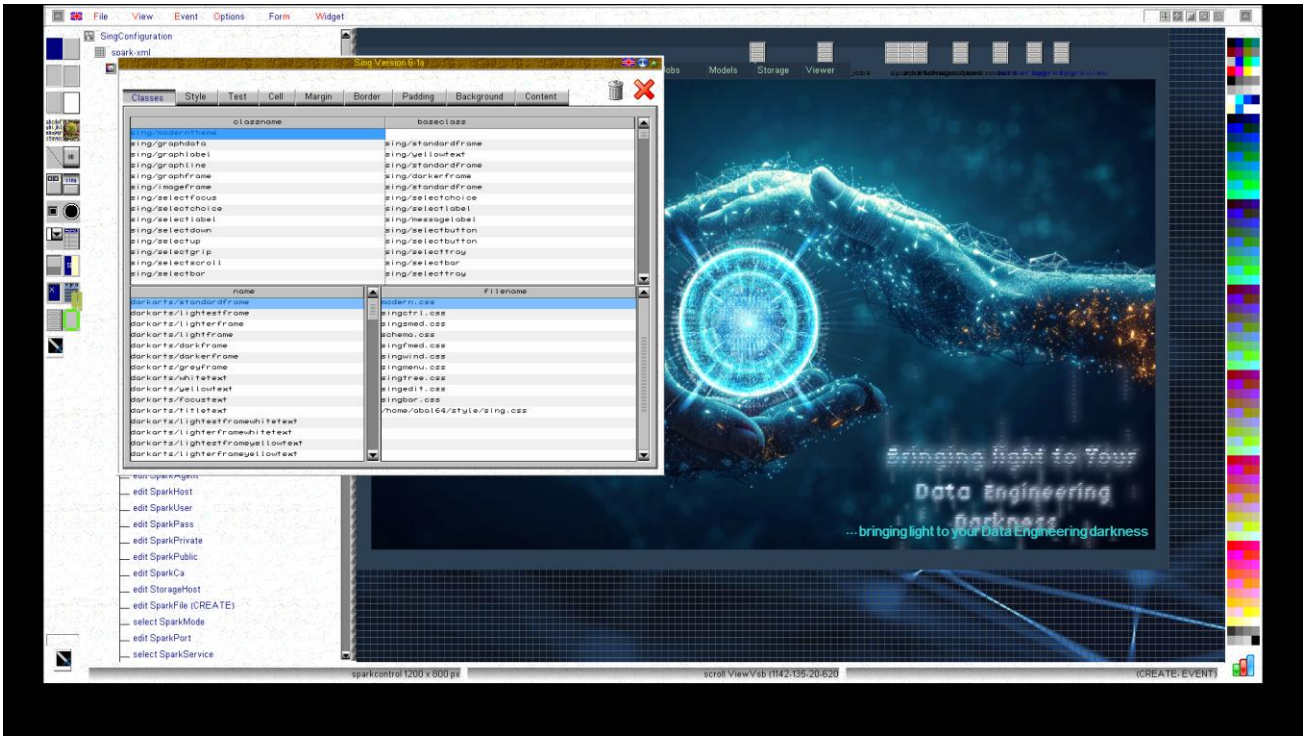
- Reset : Resets the entire database to empty.
- WS : generates a WASP Web Service endpoint for this data entity.

Other operations are currently deprecated.

STYLE EDITOR

Overview

The style editor dialog box is launched by the Top Menu option View . Style and offers nine Tab Pages for the management of style class files and definitions that can be used in the construction of the Open Abal Visual Applications.



Style Box

The Visual Style classes are similar to traditional CSS style classes and respect the same STYLE BOX model comprising the CONTEXT, MARGIN, BORDER, PADDING and CONTENT regions.

Rendering

Visual Style Classes are not Cascading since they do not describe HTML element inside HTML elements. They describe a collection of Visual Attributes which are to be applied to a provided content, text or image, at a specified default position of the indicated default dimensions.

The style class may overload the default values provided allowing the final rendering to be repositioned and re-dimensioned as required.

Deriviation

Visual Style Classes may be derived from other Style Classes and inherit (copy) the entire collection of style definitions of the parent. These properties may be modified (overloaded) as required to meet the objectives of the inheriting class and have no impact on the appearance of the original parent class.

Meta Class

Visual Style Classes may instantiate multiple Style Classes that will be presented, during rendering, with the same invocation parameters concerning position, dimensions and content. This allows a Meta Style Class to be defined which “applies” a collection of subordinate Style Classes to the received parameters by a single Visual Style operation.

Classes

This Tab Page allows selection, inspection and management of the style files and their style class definitions. The selected style class will be loaded into the Style Editor, and its properties and appearance can be consulted and modified through use of the subsequent Tab Pages and their functions.

Style

This Tab Page presents the currently selected Style Class and its associated Style File and allows a collection of self-explanatory operations to be performed.

Test

This Tab Page shows the resulting visual appearance of use of the currently select Style Class for the text content message “Test Message”.

Cell

This Tab Page allows visualisation and modifications of the CELL properties of the currently loaded Style Class.

Margin

This Tab Page allows visualisation and modifications of the MARGIN properties of the currently loaded Style Class.

Border

This Tab Page allows visualisation and modifications of the BORDER properties of the currently loaded Style Class.

Padding

This Tab Page allows visualisation and modifications of the PADDING properties of the currently loaded Style Class.

Background

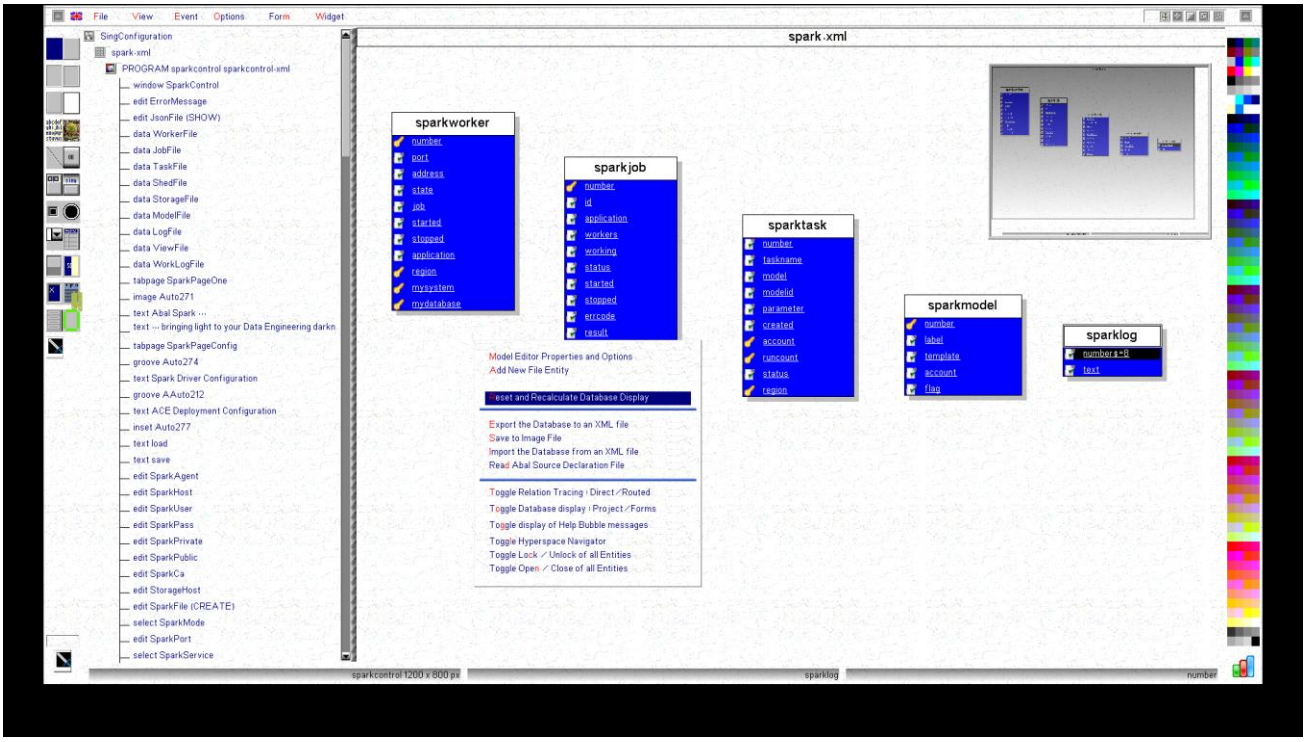
This Tab Page allows visualisation and modifications of the BACKGROUND properties of the currently loaded Style Class.

Content

This Tab Page allows visualisation and modifications of the CONTENT properties of the currently loaded Style Class.

DATABASE MODEL EDITOR

The Database Model Editor is activated from the Top Menu option View . Models . Database Model and shows the relationships between the currently defined database tables of the current form when invoked from the Forms Editor or of all database tables when invoked from the Project 3D Editor.



NAVIGATION

The Database Model may be navigated using the following MOUSE events:

- MOUSE WHEEL DOWN and UP scroll the page of the Style Classes in upward and downward directions.
- LEFT BUTTON DRAG UP / DOWN drags the page of the Style Classes in upward and downward directions.

The Database Model may be navigated using the following KEYBOARD events:

- END (CTRL B) toggles between the start and end of the list of Database Tables.
- DOWN (CTRL E) select the next Table in the list.
- UP (CTRL K) selects the preceding Table in the list.
- PAGE UP (CTRL R) moves the model display page upwards.
- PAGE DOWN (CTRL C) moves the model page downwards.
- TABULATION (CTRL I) moves the model page to the right.
- CTRL D moves the model page to the right.
- HOME (CTRL L) returns the model page view to the top left hand corner position.

- CTRL F returns control to the calling Project or Forms Editor.
- CTRL J returns the model page to left margin.
- ESCAPE returns control to the calling model Editor.

The Top Menu option File . Quit allows exit from the Database Model Editor and return to the Forms Editor or to the Project Editor.

FUNCTIONS

A RIGHT button MOUSE event in the Editor window will display the Database Model Editor contextual menu.

Table Menu

This menu will be displayed when the right button event occurs over a table header, comprising the following collection of self-explanatory options.

- File Characteristics
- Add File Member
- Recalculate Record and Index Sizes
- Copy the Current Class
- Rename the Database Model Class
- Delete the Current Class.
- Toggle Lock / Unlock State
- Toggle File Relations Display Solo / All
- Toggle Open / Close State.

Field Menu

This menu will be displayed when the right button event occurs over a table field or column, comprising the following collection of self-explanatory options.

- Define Member
- Copy Member
- Delete Member
- Add Member Before Current
- Add Member After Current
- Add Member Relation
- Toggle Primary Index Member
- Toggle Secondary Index Member
- Toggle List Column Member

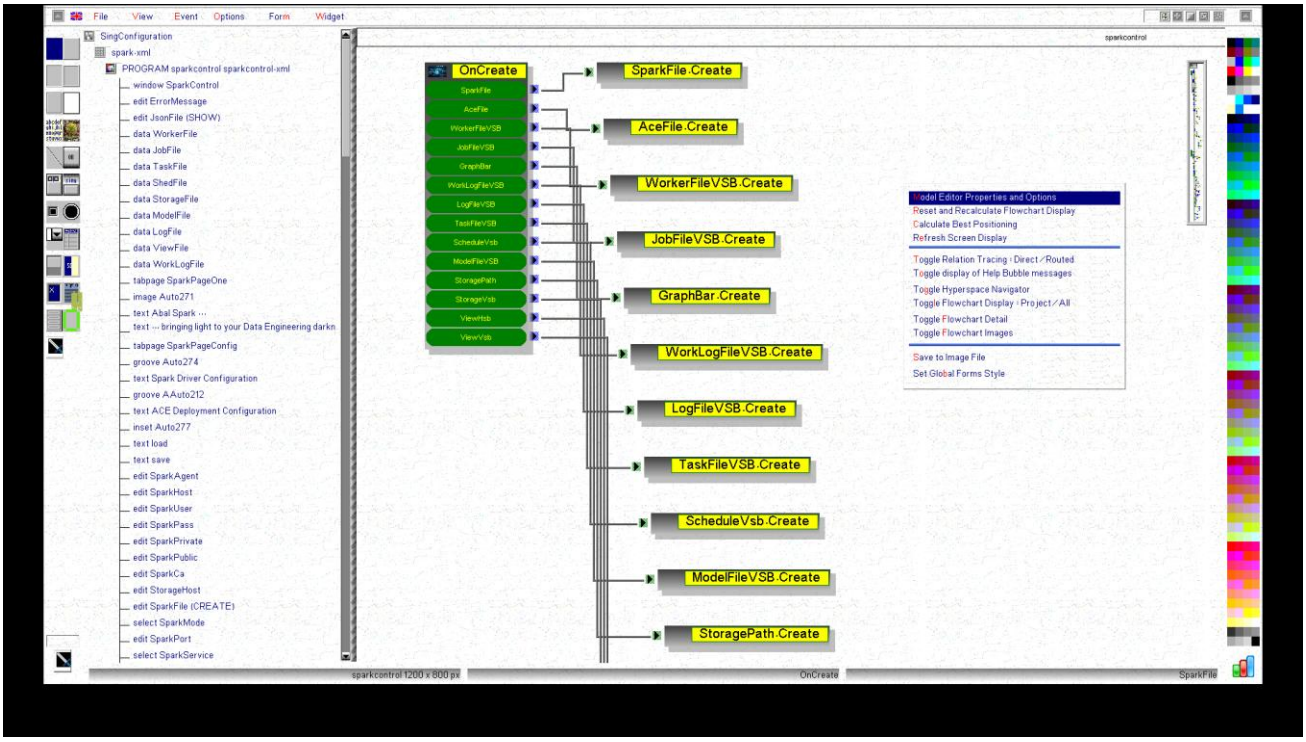
Editor Menu

This menu will be displayed when the right button event occurs elsewhere within the Editor zone, comprising the following collection of self-explanatory options.

- Model Editor Properties and Options.
- Add New File Entity.
- Reset and Recalculate Database Display
- Export the Database to an XML file.
- Save to Image File
- Import the Database from an XML File
- Read Abal Source Declaration File
- Toggle Relation Tracing : Direct / Routed
- Toggle Database Display : Project / Forms
- Toggle Display of Help Bubble Messages
- Toggle Hyperspace Navigator
- Toggle Lock / Unlock of All Entities
- Toggle Open / Close of All Entities

APPLICATION FLOW MODEL EDITOR

The Application Flow Model Editor is activated from the Top Menu option View . Models . Flow Chart and shows the relationships between the currently widgets and methods of the current Visual Form, when invoked from the Forms Editor, or the relationships between Visual Forms when invoked from the Project 3D Editor.



NAVIGATION

The Application Flow Model may be navigated using the following KEYBOARD events:

- END (CTRL B) toggles between the start and end of the list of Form Models.
- DOWN (CTRL E) select the next Form Model in the list.
- UP (CTRL K) selects the preceding Form Model in the list.
- PAGE UP (CTRL R) moves the model display page upwards.
- PAGE DOWN (CTRL C) moves the model page downwards.
- TABULATION (CTRL I) moves the model page to the right.
- CTRL D moves the model page to the right.
- HOME (CTRL L) returns the model page view to the top left hand corner position.
- CTRL F returns control to the calling Project or Forms Editor.
- CTRL J returns the model page to left margin.
- ESCAPE returns control to the calling model Editor.

The Application Flow Model may be navigated using the following MOUSE events:

- MOUSE WHEEL DOWN and UP scroll the page of the Style Classes in upward and downward directions.
- LEFT BUTTON DRAG UP / DOWN drags the page of the Style Classes in upward and downward directions.

The Top Menu option File . Quit allows exit from the Application Flow Editor and return to the Forms Editor or to the Project Editor.

FUNCTIONS

A RIGHT button MOUSE event in the Editor window will display the Database Model Editor contextual menu.

Form Menu

This menu will be displayed when the right button event occurs over a Forms Widget header, comprising the following collection of self-explanatory options.

- Inspect Form Properties
- Visual Forms Editor
- Toggle Forms Detail
- Toggle Flowchart Images
- Toggle Forms Solo
- Calculate Best Positioning

Method Menu

This menu will be displayed when the right button event occurs over a Forms Method, comprising the following collection of self-explanatory options.

- Inspect Widget Properties and Method
- Toggle Widget Method Detail
- Toggle Widget Method Solo
- Calculate Best Positioning

Editor Menu

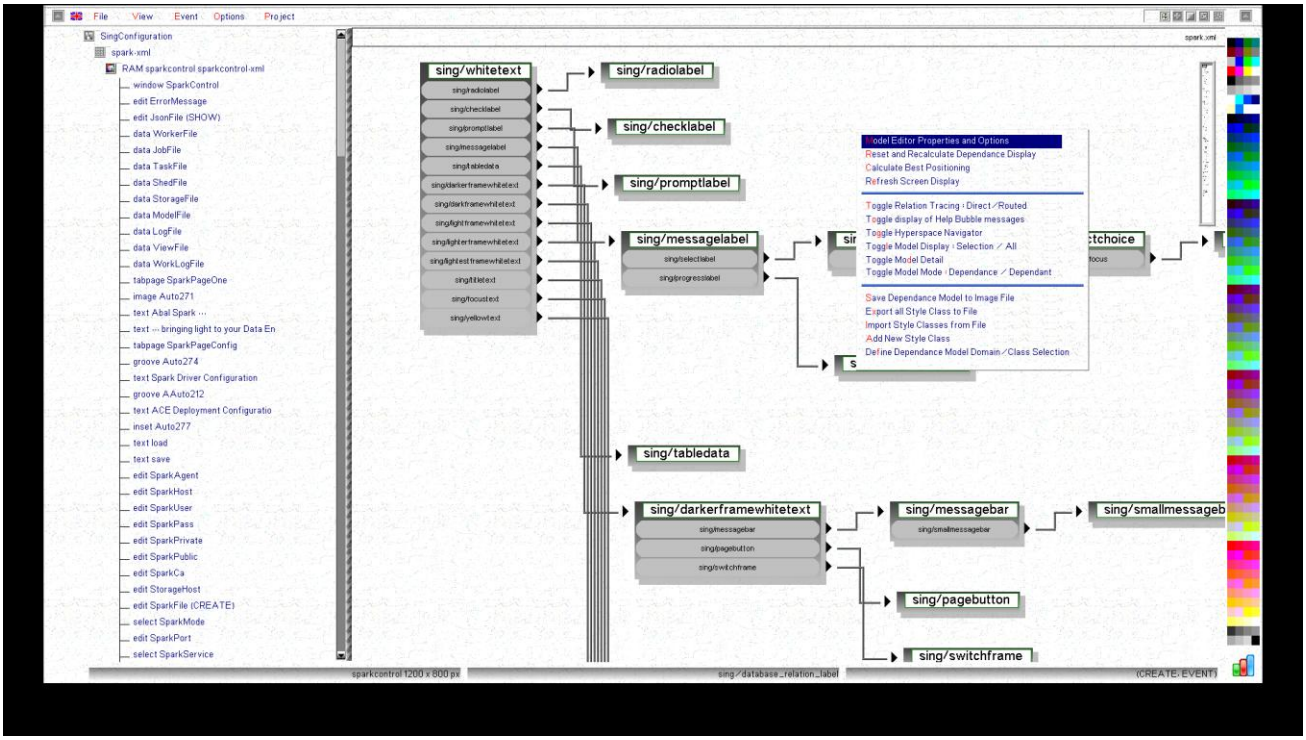
This menu will be displayed when the right button event occurs elsewhere within the Editor zone, comprising the following collection of self-explanatory options.

- Model Editor Properties and Options.
- Reset and Recalculate Database Display
- Recalculate Best Positioning
- Refresh Screen Display
- Toggle Relation Tracing : Direct / Routed
- Toggle Display of Help Bubble Messages

- Toggle Hyperspace Navigator
- Toggle Flowchart Display : Project / All
- Toggle Flowchart Detail
- Toggle Flowchart Images
- Save to Image File
- Set Global Forms Style

STYLE CLASS MODEL EDITOR

The Style Class Model Editor is activated from the Top Menu option View . Models . Style Dependence Graph and shows the relationships between the currently loaded style classes, as depicted in the following screen capture.



NAVIGATION

The Style Class Model may be navigated using the following MOUSE events:

- MOUSE WHEEL DOWN and UP scroll the page of the Style Classes in upward and downward directions.
- LEFT BUTTON DRAG UP / DOWN drags the page of the Style Classes in upward and downward directions.

The Style Classes Model may be navigated using the following KEYBOARD events:

- END (CTRL B) toggles between the start and end of the list of Style Classes.
- DOWN (CTRL E) select the next Style Class in the list.
- UP (CTRL K) selects the preceding Style Class in the list.
- PAGE UP (CTRL R) moves the model display page upwards.
- PAGE DOWN (CTRL C) moves the model page downwards.
- TABULATION (CTRL I) moves the model page to the right.
- CTRL D moves the model page to the right.
- HOME (CTRL L) returns the model page view to the top left hand corner position.
- CTRL F returns control to the calling Project or Forms Editor.

- CTRL J returns the model page to left margin.
- ESCAPE returns control to the calling model Editor.

The Top Menu option File . Quit allows exit from the Style Class Model Editor and return to the Forms Editor or to the Project Editor.

FUNCTIONS

A RIGHT button MOUSE event in the Editor window will display the Style Class Model Editor contextual menu.

Class Menu

This menu will be displayed when the right button event occurs over a Style Class header, comprising the following collection of self-explanatory options.

- Inspect Style Class Instructions
- Inspect Effective Style Class
- Export Style Class to File
- Toggle Style Class Solo
- Toggle Style Class Details
- Add Class Relation
- Calculate Best Positioning

Editor Menu

This menu will be displayed when the right button event occurs elsewhere within the Editor zone, comprising the following collection of self-explanatory options.

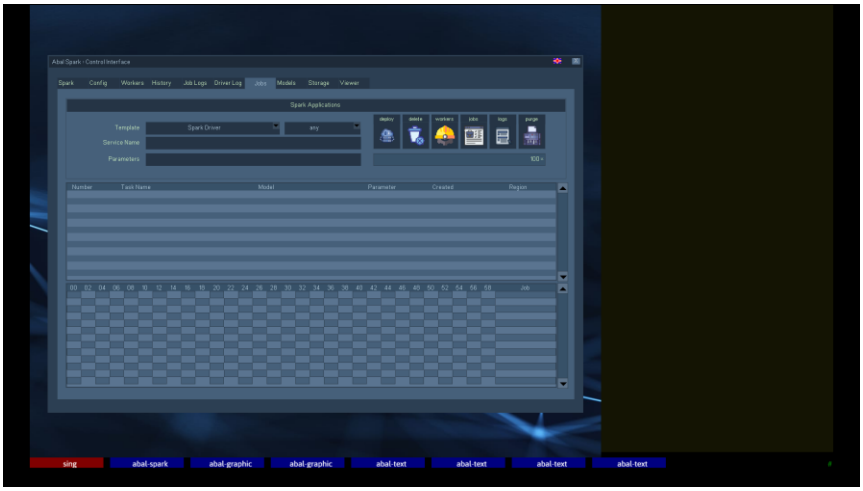
- Model Editor Properties and Options.
- Reset and Recalculate Dependence Display
- Recalculate Best Positioning
- Refresh Screen Display
- Toggle Relation Tracing : Direct / Routed
- Toggle Display of Help Bubble Messages
- Toggle Hyperspace Navigator
- Toggle Model Display : Selection / All
- Toggle Model Detail
- Toggle Model Display : Dependence / Dependents
- Save Dependence Model to Image File
- Export All Style Classes to File
- Import Style Classes From File
- Add New Style Class
- Define Dependence Model Domain / Class Selection

USING STYLE

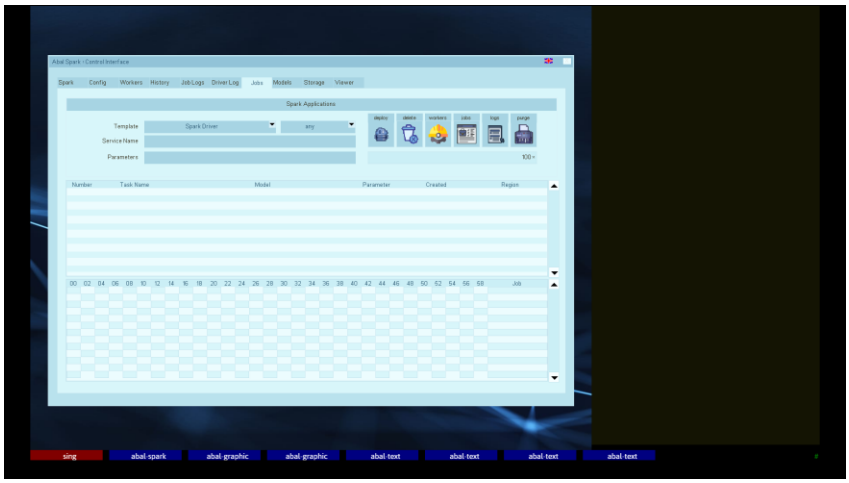
The widgets of the Visual Library were originally programmed for a WINDOWS 95 relief-based style. Visual programming presentation tastes, in general, have greatly progressed since those early days, following the fashion of the day.

To this effect SING allows the default widget representation to be overloaded by a collection of style classes corresponding to each of the aspects of those widgets. An example of this can be seen below where the style sheets Dark Arts and Day Light, both variations of the Modern foundation classes, are switched from inside the application.

Dark Arts Style Sheet



Day Light Style Sheet



For further information relating to the style properties that are supported by the style processor of the Visual Library please refer to the section of this document entitled Visual Style Properties.

The following collection of modern style classes of varying hues and sizes is currently distributed with SING and may be used as a basis for the creation of your own.

- True Blue 2 and 3

- Baby Blue 2 and 3
- Warm Blue 2 and 3
- Earthbound 2 and 3
- Nature 2 and 3
- True Green 2 and 3
- Dark Arts 2 and 3
- Day Light 2 and 3

The versions 2 and 3 correspond to the use of the ARIAL2 and ARIAL3 fonts which require small edit and large edit fonts to be used respectively for Vusal Table widget dimensioning.

Each of the above styled themes make use of the common class definition file:

- Modern.css

The definition of each and any of the following list of classes allows the visual libraries to be restyled according to your specific tastes.

Visual Frames

The following style classes may be defined and included for the overloading of the four basic frame types.

- Outsetframe
- Insetframe
- Grooveframe
- Ridgeframe

Visual Window

The following classes may be defined and included for the overloading of the visual window widget elements.

- Windowframe
- Windowtitle
- Windowexit
- Windowhelp
- Windowmaxi
- Windowmini

Visual Tab Page

The following classes may be defined and included for the overloading of the visual tab page widget elements.

- Pageframe

- Pagebutton
- Pagefocus

Visual Table

The following classes may be defined and included for the overloading of the visual table widget elements.

- Tabletray
- Tableframe
- Tableinner
- Tableheader
- Tablerow
- Tablecolumn
- Tablecell
- Tabledata
- Tablefocus

The font size parameter passed to the visual table widget instruction, in SING, will be used to condition the dimensioning of the support variables for the management of the data to be tabulated and should correspond to the font dimensions used in the subsequent overloading classes.

Visual Scroll

The following classes may be defined and included for the overloading of the visual scroll widget elements.

- vscrollframe
- vscrollbar
- scrollup
- scrolldown
- hscrollframe
- hscrollbar
- scrollleft
- scrollright

Visual Edit

The following classes may be defined and included for the overloading of the visual edit widget elements.

- editframe
- editfocus
- editshow

- editgetfocus
- editlosefocus

Visual Button

The following classes may be defined and included for the overloading of the visual push button widget elements.

- Buttonframe
- Iconframe
- Buttonfocus
- Iconfocus
- Iconbutton

Visual Switch

The following classes may be defined and included for the overloading of the visual switch widget elements.

- Switchframe
- Switchtrue

Visual Check

The following classes may be defined and included for the overloading of the visual check box widget elements.

- checkframe
- checktrue
- checklabel

Visual Radio

The following classes may be defined and included for the overloading of the visual radio button widget elements.

- Radioframe
- Radiotrue
- Radiolabel

Visual Progress

The following classes may be defined and included for the overloading of the visual progress bar widget elements.

- progressframe
- progressfill

- progresswipe
- progresslabel

Visual Select

The following classes may be defined and included for the overloading of the visual selection list widget elements.

- Selectframe
- Selecttray
- Selectbar
- Selectscroll
- Selectgrip
- Selectup
- Selectdown
- Selectlabel
- selectchoice
- Selectfocus

Visual Graph

The following classes may be defined and included for the overloading of the visual data graph widget elements.

- Graphframe
- Graphline
- Graphlabel
- Graphdata

Visual Text

These widgets cannot be overloaded globally and should have their own application specific style class corresponding to the semantics of use of the text.

Visual Image

These widgets cannot be overloaded globally and should have their own application specific style class corresponding to the semantics of use of the text.

Visual Line

These widgets cannot be overloaded globally or locally.

Visual Fill

These widgets cannot be overloaded globally or locally.

Visual Style Properties

Introduction

Visual Style Classes allow Open Abal Applications to be designed based on modern dynamic styling techniques where Style Class names are associated with the elements comprising the Graphical User Interface. The actual appearance of the Visual Element will depend on the definition or composition of the involved Style Classes.

Style Class Files allow the definition of a collection of Style Classes that can be loaded together.

Style Class definition follows the traditional syntax introduced by CSS1 and maintained with the subsequent versions CSS2 and CSS3.

A Named Style Class definition comprises a semi-colon separated collection of Style Clauses contained between the traditional left and right curly braces. Style clauses comprise a clause name and a clause value, separated by a colon character.

Unquoted white space is silently discarded both inside and around Style Class definitions.

```
Classname { clausename: value; clausename: value }
```

A collection of Style Class definitions may be regrouped into Style Class Domains allowing identical Style Class names to be defined with differing Style Clauses and subsequent appearance.

The Style Class Domain can be specified by the Visual Application subsequently selecting only the Style classes that were defined under the corresponding Domain.

```
@domain one;
Classname { clausename: value; clausename: value }
@domain two;
Classname { clausename: value; clausename: othervalue }
```

The visual style properties of the Visual Library Style processor address the standard elements of the style box structure as shown below:

- Special
- General
- Position
- Dimensions
- Margin
- Border
- Padding
- Content

Each of these elements of the style box structure will be discussed in detail below.

Special

- **@domain** : this clause of a Style Class definition indicates that the subsequent term will be used as the name of the Style Domain to which the subsequent Style Class definitions will be added.
- **@define** : this clause of a Style Class definition indicates that the subsequent term will be defined as the remaining text up until the clause terminating semi-colon. This allows terms to be defined for dynamic substitution in subsequent style classes and style class clauses.
- **@import** : this clause of a Style Class definition indicates that the subsequent URL term will be expected to be the name of a Style Class file to be loaded and imported using the current STYLEPATH configuration.

General

- **Class** : this clause of a Style Class definition indicates that the subsequent term defines the name of the Style Class from which this Style class is derived. This clause will cause all previous clauses to be in-effective since the contents of the parent Style Class will be copied to the Style Definition Container of the inheriting class.
- **Alias** : this clause of a Style Class definition indicates that the subsequent term defines the name of a Style Class to be instantiated for the same eventual invocation properties.

Colour

In the following style clause descriptions, whenever reference is made to a colour value, it may be described in one of the following ways:

- black
- navy
- green
- cyan
- red
- brown
- magenta
- grey
- silver
- blue
- lime
- sky
- pink
- yellow
- purple

- white
- ihm : this keyword is followed by a left and right bracketed integer between 0 and 15 representing a colour in the IHM (32 to 47) of the current colour mapped palette.
- rgb : this keyword is followed by a left and right bracketed collection of three colour values between 0 and 255 representing the red, green and blue colour portions.
- color: this keyword is followed by a left and right bracketed integer between 0 and 255 representing a colour of the current colour mapped palette.
- A "#" prefix followed by a sequence of six hexa-decimal digits representing in order the red, green add blue portion of the true RGB colour.

Colour

In the following style clause descriptions, whenever reference is made to a value, it may be an integer or floating-point value suffixed with one of the following qualifiers:

- px : the value is expressed in pixels
- pt : the value is expressed in points
- mm : the value is expressed in millimetres
- cm : the value is expressed in centimetres
- in: the value is expressed in inches.
- % : the value is expressed as a percentage value.

```
example
{
  left:   3px;
  right:  2mm;
  bottom: 1cm;
  right:  0.5in;
}
```

Align

In the following style clause descriptions, whenever reference is made to an alignment value, it may be any of the following alignment types according to the corresponding usage:

- none
- left
- right
- center
- justify
- repeat
- tile
- makefit

- bestfit
- transparent

```
example
{
  text-align:     center;
  background-align:  makefit;
}
```

Position

The position properties and modifiers determine the positioning of the style box on the output canvas or page.

Position

- **home** : this style clause establishes the positioning relative to the top left corner.
- **static** : this style clause requires the invocation position to be respected.
- **auto** : this style clause requires the invocation position to be respected.
- **relative :** this style clause indicates that the values provided by the left, right, top and bottom clauses will be relative to the style invocation coordinates.
- **absolute** : this style clause indicates that the values provided by the left, right, top and bottom clauses will be absolute screen coordinates.
- **fixed** : this style clause indicates that the values provided by the left, right, top and bottom clauses will be absolute screen coordinates.

Left

This style clause establishes the left margin position with respect to the value of the position clause.

Right

This style clause establishes the right margin position with respect to the value of the position clause.

Top

This style clause establishes the top margin position with respect to the value of the position clause.

Bottom

This style clause establishes the bottom margin position with respect to the value of the position clause.

Dimensions

The dimensions properties and modifiers determine the overall occupation of the style box on the output canvas or page.

Width

This style clause establishes the width of the final style box overloading the value provided by each style invocation.

Height

This style clause establishes the height of the final style box overloading the value provided by each style invocation.

Margin

The margin properties determine the existence and appearance of the margin that is situated between the position of the style box and its border.

The following extensions to the margin keyword are available and will be described in detail in the following paragraphs.

- width
- color
- gradient
- top
- right
- bottom
- left
- align
- image

Margin-Image

This clause sets the image to be used as the background of the margin region of the style cell.

```
Example { margin-image: url(test.png); }
```

Margin-Color

This clause sets the colour of the margin region of the style cell. The clause may be prepended with left, right, top and bottom extensions allowing each border region colour to be individually addressed.

```
example
{
  Margin-top-color:    red;
  Margin-right-color:  blue;
```

```
  Margin-bottom-color: green;
  Margin-left-color:   white;
}
```

Margin-Width

This clause sets the size of the margin region of the style cell. The clause may be prepended with left, right, top and bottom extensions allowing each border region size to be individually addressed.

```
example
{
  Margin-top-width:    4px;
  Margin-right-width:  2px;
  Margin-bottom-width: 3px;
  Margin-left-width:   1px;
}
```

Margin-Align

This clause sets the image alignment of the margin region of the style cell. The clause may be prepended with left, right, top and bottom extensions allowing each margin region alignment to be individually addressed.

```
example
{
  Margin-top-align:    tile;
  Margin-right-align:  center;
  Margin-bottom-align: tile;
  Margin-left-align:   center;
}
```

Border

The border properties determine the existence and appearance of the style box border, situated between the margin and the padding.

The following extensions to the border keyword are available and will be described in detail in the following paragraphs.

- width
- style
- color
- gradient
- top

- right
- bottom
- left
- align

Border-Image

This clause sets the image to be used as the background of the border region of the style cell.

```
Example { border-image: url(test.png); }
```

Border-Style

This clause sets the style and presence of the border region of the style cell. The clause may be appended with left, right, top and bottom extensions allowing each border region size to be individually addressed. The following border variations are possible.

- **none** : the border region will be disactivated and will not participate in the style bounding box calculations.
- **solid** : the border will be draw with a square cornered **solid** single **line** of the color specified by **border-color** and width specified by **border-size**.
- **double :** the border will be draw with a **solid** double **line** of the color specified by **border-color** and width specified by **border-size**.
- **inset** : the border will be draw with the **inset relief** colours and the width specified by **border-size**.
- **outset** : the border will be draw with the **outset relief** colours and the width specified by **border-size**.
- **groove** : the border will be draw with the **groove relief** colours and the width specified by **border-size**.
- **ridge** : the border will be draw with the **ridge relief** colours and the width specified by **border-size**.
- **url :** the border will be draw using the image retrieved from the URL and the width specified by **border-size**.
- **rounded** : the border will be drawn with a rounded cornered **solid** single **line** of the color specified by **border-color** and width specified by **border-size**.
- **disk** : the border will be drawn with a circular **solid** fill of the color specified by **border-color** and width specified by **border-size**.
- **circle** : the border will be drawn with a circle **solid** single **line** of the color specified by **border-color** and width specified by **border-size**.
- **shade** : the border area will be filled with a shaded color specified by **border-color** and width specified by **border-size**.
- **hole** : the border area outside of than the circular central disk area, will be filled with a color specified by **border-color** and width specified by **border-size**.
- **edit** : a standard edit coloured frame will be drawn.

- **concave** : a concave relief colouring will be drawn for the entire border area.
- **convex** : a convex relief colouring will be drawn for the entire border area.
- **dashed** : the border will be draw with a square cornered **dashed** single **line** of the color specified by **border-color** and width specified by **border-size**.
- **dotted** : the border will be draw with a square cornered **dotted** single **line** of the color specified by **border-color** and width specified by **border-size**.
- **vconvex** : a vertical convex colouring using foreground and background colours will be drawn for the entire border area.
- **hconvex** : a horizontal convex colouring using foreground and background colours will be drawn for the entire border area.
- **hconcave** : a vertical concave colouring using foreground and background colours will be drawn for the entire border area.
- **vconcave** : a horizontal concave colouring using foreground and background colours will be drawn for the entire border area.
- **vgradient** : a vertical gradient colouring between foreground and background colours will be drawn for the entire border area.
- **hgradient** : a horizontal gradient colouring between foreground and background colours will be drawn for the entire border area.

```
example
{
  Border-top-style:    solid;
  Border-right-style:  dotted;
  Border-bottom-style: double;
  Border-left-style:   dashed;
}
```

Border-Color

This clause sets the colour of the border region of the style cell. The clause may be appended with left, right, top and bottom extensions allowing each border region size to be individually addressed.

```
example
{
  Border-top-color:    red;
  Border-right-color:  blue;
  Border-bottom-color: green;
  Border-left-color:   white;
}
```

Border-Gradient

This clause sets the final colour of the border region of the style cell to which the gradient will converge. The clause may be appended with left, right, top and bottom extensions allowing each border region size to be individually addressed.

```
example
{
  Border-top-gradient:    white;
  Border-right-gradient:  white;
  Border-bottom-gradient: white;
  Border-left-gradient:   white;
}
```

Border-Width

This clause sets the size of the border region of the style cell. The clause may be appended with left, right, top and bottom extensions allowing each border region size to be individually addressed.

```
example
{
  Border-top-width:    4px;
  Border-right-width:  2px;
  Border-bottom-width: 3px;
  Border-left-width:   1px;
}
```

Border-Align

This clause sets the image alignment of the border region of the style cell. The clause may be prepended with left, right, top and bottom extensions allowing each border region alignment to be individually addressed.

```
example
{
  Border-top-align:    tile;
  Border-right-align:  center;
  Border-bottom-align: tile;
  Border-left-align:   center;
}
```

Corner

This clause of a Style Class definition describes the styling of corners of the border region of the Style Cell. The following extensions may be used with this clause.

- Top
- Bottom

Corner-Top

This clause sets the image to be used as the top corner of the border region of the style cell. The extensions left and right are required to correctly identify the specific corner.

```
example
{
  corner-top-left: url(cornertl.png);
  corner-top-right: url(cornertr.png);
}
```

Corner-Bottom

This clause sets the image to be used as the bottom corner of the border region of the style cell. The extensions left and right are required to correctly identify the specific corner.

```
example
{
  corner-bottom-left: url(cornerbl.png);
  corner-bottom-right: url(cornerbr.png);
}
```

Padding

The padding properties determine the existence and appearance of the style box padding, situated between the border region and the content. The clause may be appended with left, right, top and bottom extensions allowing each border region size to be individually addressed.

```
example
{
  Padding-top:    1px;
  Padding-right:  1mm;
  Padding-bottom: 1cm;
  Padding-left:   1in;
}
```

Content

The content properties determine the existence and appearance of the style box content payload, situated inside of the padding.

Letter Spacing

This clause of a Style Class definition allows definition of a value to be used as the inter letter spacing for all textual content.

```
example { letter-spacing: 2px; }
```

Word Spacing

This clause of a Style Class definition allows definition of a value to be used as the word spacing with TEXT label and trigger content when space reduction is performed.

```
example { word-spacing: 2mm; }
```

Line Spacing

This clause of a Style Class definition allows definition of a value to be used as the inter line spacing with multi line TEXT, EDIT and TRIGGER content.

```
example { line-spacing: 1cm; }
```

Content

This clause of a Style Class definition describes the nature of the content for which this class is intended to be applied. This affects the way in which the content string provided by the invocation through Visual Style is interpreted.

- **URL** : the content string will be interpreted as a URL value describing an IMAGE to be displayed as content.
- **EDIT** : the content string will be displayed as a value for an EDIT FIELD with no space reduction applied.
- **TEXT** : the content string will be displayed as decorative text or label with eventual space reduction applied.
- **TRIGGER** : the content string will be displayed as trigger text or label with trigger character detection and highlight, and with eventual space reduction applied.
- **GRIP** : the content string will be ignored and overload by the standard Scroll Bar GRIP image.
- **UP** : the content string will be ignored and overload by the standard Up Button image.
- **DOWN** : the content string will be ignored and overload by the standard Down Button image.
- **TOP** : the content string will be ignored and overload by the standard Top Button image.
- **BOTTOM** : the content string will be ignored and overload by the standard Bottom Button image.
- **LEFT** : the content string will be ignored and overload by the standard Left Button image.
- **RIGHT** : the content string will be ignored and overload by the standard Right Button image.

- **AUTO** : no explicit content indication is provided.
- **NONE** : the content string will be simply ignored, and no content will be displayed.

Text

This clause of a Style Class definition describes the nature of textual content in terms of its colour, alignment and the text font characteristics.

Text-color

This clause of a Style Class definition allows definition of a colour value to be used as foreground text colour for use with TEXT, EDIT and TRIGGER content.

```
example { text-color: red; }
```

Text-align

This clause of a Style Class definition allows definition of an alignment value to be used as foreground text colour for use with TEXT, EDIT and TRIGGER content.

```
example { text-align: left; }
```

Text-font

This clause of a Style Class definition allows definition of a text font to be used as foreground text colour for use with TEXT, EDIT and TRIGGER content.

```
example { text-font: url(arial3.fmf); }
```

Text-style

This clause of a Style Class definition allows definition of a text attribute value to be used as foreground text colour for use with TEXT, EDIT and TRIGGER content. The following text decoration attribute values are available:

- none
- bold
- underline
- overline
- line-through
- blink
- shadow

```
example { text-style: bold; }
```

Text-decoration

This clause of a Style Class definition allows definition of a text attribute value to be used as foreground text colour for use with TEXT, EDIT and TRIGGER content. The following text decoration attribute values are available:

- none

- bold
- underline
- overline
- line-through
- blink
- shadow

```
example { text-decoration: underline; }
```

Text-indent

This clause of a Style Class definition allows definition of a text indent value to be used as foreground text colour for use with TEXT, EDIT and TRIGGER content.

```
example { text-indent: 1cm; }
```

Background

This clause of a Style Class definition describes the nature of the background behind the content for which this class is intended to be applied. The following extensions may be used with this clause.

- Color
- Gradient
- Image
- Attach
- Align

Background-Color

This clause sets the colour of the background of the content region of the style cell.

```
example { background-color: blue; }
```

Background-Gradient

This clause sets the final colour of the background colour gradient of the content region of the style cell.

```
example { background-color: red; background-gradient: blue; }
```

Background-Image

This clause sets the URL of the image to be used to cover the background of the content region of the style cell.

```
example { background-image: url(picture.png); }
```

Background-Attach

This clause sets the image attachment value of the background image to be used to cover content region of the style cell. This clause may specifiy one of the following values:

- fixed : the background will be fixed
- scroll : the background will scroll with the content.

```
example { background-attach: scroll; }
```

Background-Align

This clause sets the image alignment value of the background image to be used to cover the content region of the style cell.

```
example { background-align: makefit; }
```

Shadow

This clause of a Style Class definition describes the application of shadow under the Style Cell. The following extensions may be used with this clause.

- Color
- Width
- Height

Shadow-color

This clause sets the colour of the Style Cell Shadow.

```
example { shadow-color: grey; }
```

Shadow-width

This clause sets the width of the Style Cell Shadow.

```
example { shadow-width: 2mm; }
```

Shadow-height

This clause sets the height of the Style Cell Shadow.

```
example { shadow-height: 3mm; }
```

SING GETTING STARTED

The following simple workflow allows you to create your first SING project and Forms Model.

1. Launch the SING tool for your environment and the project example.
2. Select the Top Menu option Project. Add Form to access the New Form Creation dialog box.
3. Specify the name “newform”
4. Click the green button to create the basic form.
5. The grey 3D object is now visible in the 3D model editor.
6. Select the Top Menu option View. Visual to launch the Forms Editor.
7. Arrival in the Forms Editor.
8. Select the WINDOW tool in the Tool Bar
9. Press the left button of the MOUSE and drag to create a window of the size you desire.
10. Release the left button and the WINDOW will appear.
11. Select the PUSH BUTTON tool in the Tool Bar
12. Press the left button of the MOUSE and drag to create a push button of the size you desire.
13. Release the left button and the PUSH BUTTON will appear.
14. Select the Top Menu option View. Properties to launch the Widget Properties dialog box.
15. Change the value of the text of the PAYLOAD field to “Hello World”
16. Click the green button to validate modification of the Widget Properties.
17. Select the Top Menu option File. Save to save the current state of your Forms Model.
18. Select the Top Menu option File. Production to launch the Widget Properties dialog box.
19. Click the green button to validate the launch of the Forms Editor Production.
20. Select the Top Menu option View. Run to launch the resulting OPENABAL program.
21. The visual window and push button will appear on the screen.
22. Press ESCAPE to return to SING
23. Select the Top Menu option View. Application to return to the Project Editor.
24. Select the Top Menu option File. Save to save the current state of your Project or Application Model.
25. Select the Top Menu option File. Quit to leave SING.

Congratulations you have created your first visual program using SING.

ENVIRONMENT VARIABLES

The following environment variables may be set to configure the corresponding behaviour of SING and the VISUAL library used by generated OPENABAL programs.

Name	Value	Description
SINGSTYLE	Style filename	Specifies the default style for the display of all user interface constructions in the SING domain.
SINGXML	DTD filename.	Specifies the name of the DTD for SING XML loading. This was traditionally threed.dtd for the Project Editor and sing.dtd for the Forms Editor. This is no longer required.
SINGPERM	0 or 1 *	Activate use of Windows Permissions control.
STYLEPATH	System path	Specifies the default visual style path. This is a fall-back value since the value should be loaded from the configuration.
PALETTE	Palette file name and path	Specifies the default visual palette filename. This is a fall-back value since the value should be loaded from the configuration.
COMPONENTPATH	System path	Specifies the default visual component path. This is a fall-back value since the value should be loaded from the configuration.
GIGO	Mode:Font:Palette:Colour	Specifies the default visual graphics configuration. **Mode** A VESA mode(between 0 and 7) or and '=' prefixed, comma separated graphical description comprising width, height. **Font** The default Text Font **Palette** The default RGB colour map palette.

		Colour Three hexadecimal digits between 0 and F describing default foreground, background and erase colours.
GIGOTITLECOLORON	0 * or 1	Activates use of alternative colours for standard title font and background colours (when no style used). **Default** FG=x1C, BG=x1B **Alternative** FG=x19, BG=x1A
LTSGIGO	0 * or 1	Specifies the use of the LTS graphics interface for GIGO graphics. By default, the X11 interface is used.
VISUALTRACE	0 * or 1	Activates tracing of VISUAL library function calls, by the VISUAL library, during OPEN ABAL program execution
VISUALFREEZE	0 or 1 *	Controls the behaviour of the VISUAL FREEZE instruction. 0 : nothing 1 : EVENTS 300 and 301
STYLETRACE	0 or 1 *	Specifies use of ERROR tracing when style errors are encountered.
ECHOSTYLE	0 * or 1	Activates echo to standard output of style instructions during style class parsing.
XMLECHO	0 * or 1	Activates echo to standard output of XML during PROJECT and FORMS parsing.
ABALECHO	0 * or 1	Activates echo to standard output of ABAL source instructions during ABAL source import operations.
VWIDGET	0 * or 1	Specifies the nature of the Widget Properties dialog box as: 0 : widget type specific

		1: generic for all widgets.
WIDGETRESIZE	0 * or 1	Specifies how to resize visual widgets in the Forms Editor: 0 : LEFT button on widget frame or edge. 1 : RIGHT button inside widget.
SINGFORMSORT	0 * or 1	Specifies generation of Forms SORT methods.
NOASB	0 * or 1	Inhibits the use of ASB for post generation ABAL source beautification.
LDAP CONFIGURATION OPTIONS		
ABALHOST	Domain name or IP	Specifies the default LDAP host IP address or domain name.
LDAPROOT	String	Specifies the ROOT of the LDAP configuration section.
VLDAPHOST	Domain name or IP	Specifies the default LDAP host IP address or domain name.
VLDAPUSER	Username string	Specifies the USERNAME credential for LDAP access.
VLDAPACCESS	User password string	Specifies the PASSWORD credential for LDAP access.