Neil J Martin
neilm@4js.com

Genero User Interface
Dynamic User Interface - DUI

Genero BDL 3.10
Genero Studio 3.10

FOUR Js
The Power of Simplicity

January 2018

Goals

Describe and manipulate the first layer of the new Dynamic User Interface

Create Action Defaults

Use Presentation Styles

Create XML files for a ToolBar or TopMenu

Create a Start Menu

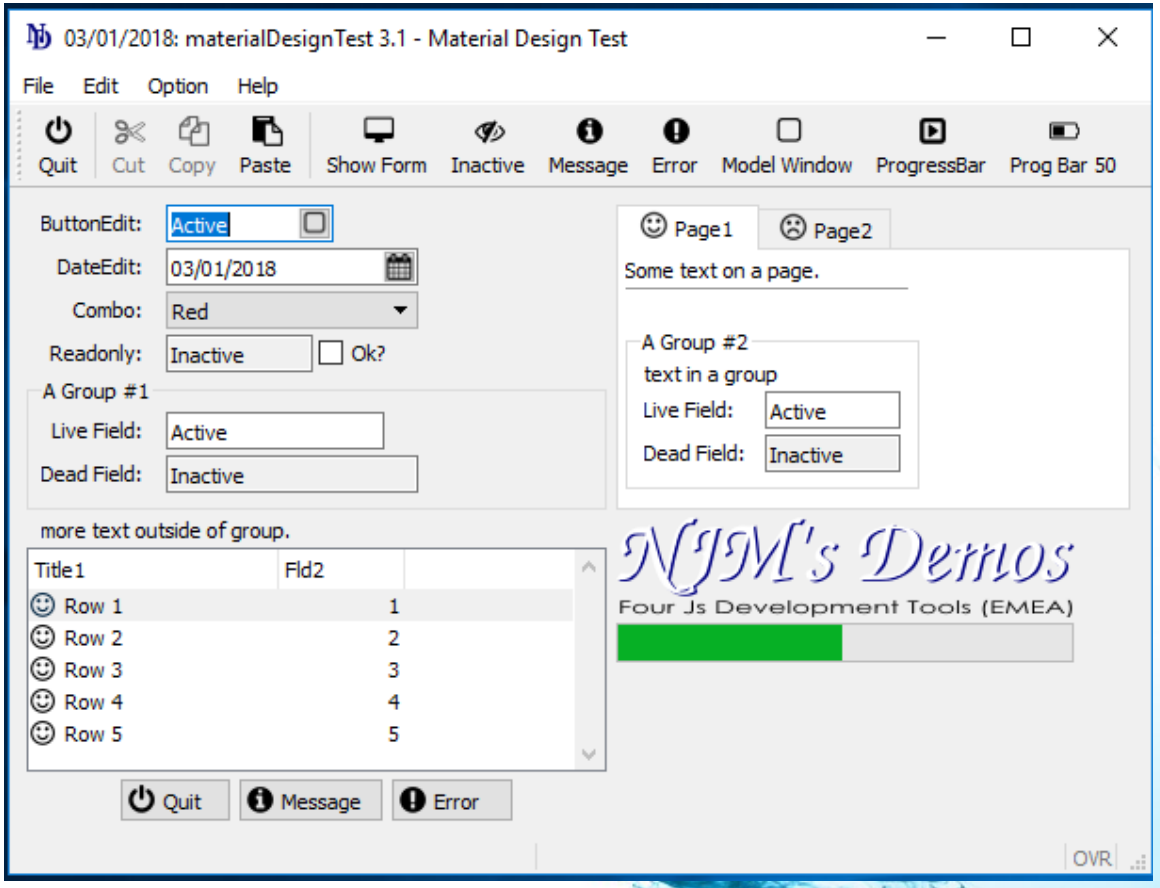
Use MDI Windows

Load XML files dynamically at runtime

DUI – Top Level

The first layer of the Dynamic User Interface.

- Windows and Forms
- Presentation Styles
- Statusbar
- Ring Menus
- Actions
- Toolbars
- Top Menus
- Start Menus



FOUR Js
The Power of Simplicity

DUI – Concepts

Windows are created from programs as independent resizable windows.

Windows define a display context and layout for sub-elements like forms, menus, message and error lines, …

A window can contain only one **Form** at a time.

You can apply independent **window styles** to each window.

You can use the **Window** class to manipulate window objects.

The **Form** class can be used to handle forms.

DUI – Concepts

Windows are graphical objects that react in the same way as the default behavior of the Client environment.

For example, in GDC for Windows, a user can close the current window by pressing Alt-F4.
This will send a cancellation ACTION to the DVM.
It is up to the program to manage it with:

```
ON ACTION close
```

There is also:

```
OPTION ON TERMINATE SIGNAL
```

FOUR Js
The Power of Simplicity

Presentation Styles

FOUR Js
The Power of Simplicity

DUI – Presentation Styles

Presentation Styles are provided to centralize attributes related to the appearance of user interface elements.

Elements in the AUI that reference a Style attribute will automatically get the attribute values defined in that style.

Elements in the AUI that don't reference a style will use the global default style.

Presentation styles are defined in a unique file with the extension .4st. Genero provides a default style file, $FGLDIR/lib/default.4st.

You can create your own style file to replace default.4st.

DUI – Presentation Styles - Example .4st file

```
<StyleList>
  <Style name="Window">
     <StyleAttribute name="windowType" value="normal" />
  </Style>
  <Style name="Window.main">
     <StyleAttribute name="windowType" value="normal" />
     <StyleAttribute name="startMenuPosition" value="menu" />
  </Style>
  <Style name="Window.main2">
     <StyleAttribute name="windowType" value="normal" />
     <StyleAttribute name="actionPanelPosition" value="none" />
     <StyleAttribute name="ringMenuPosition" value="none" />
  </Style>
  <Style name="Window.dialog">
     <StyleAttribute name="windowType" value="modal" />
     <StyleAttribute name="sizable" value="no" />
     <StyleAttribute name="position" value="center" />
     <StyleAttribute name="actionPanelPosition" value="bottom" />
     <StyleAttribute name="ringMenuPosition" value="bottom" />
     <StyleAttribute name="toolBarPosition" value="none" />
     <StyleAttribute name="statusBarType" value="none" />
     <StyleAttribute name="errorMessagePosition" value="popup" />
  </Style>
```

DUI – Presentation Styles - Syntax

```
<StyleList>
  <Style name="style-identifier" >
     <StyleAttribute name="attribute-name"
value="attribute-value" />
     [...]
  </Style>
  [...]
</StyleList>
```

style-identifier is the name of the style referenced by graphical objects with the Style attribute.
attribute-name defines the name of the graphical object attribute.
attribute-value defines the value to be assigned to attribute-name.

DUI – Presentation Styles

Styles can be defined at several levels:

`<Style name="*">`

Global; applied automatically to all elements

`<Style name="element-type">`

Global; applied automatically to the named element type (Window, Edit, Combobox, …)

`<Style name=".any-name">`

Specific; applied to any element types that have the Style attribute set to this Style name.

`<Style name="element-type.any-name">`

Specific; applied only to elements of the given type that have the Style attribute set to this Style name.

DUI – Presentation Styles - Defaults

Dialog : for typical modal windows like winmessage or winquestion

Naked : No Menu, no Action Buttons

Main : Start Application Window, ideal to display a StartMenu

If you don't specify a STYLE, there is a default Style defined as 'Window'.

FOUR Js
The Power of Simplicity

DUI – Presentation Styles - Defaults

Use the attribute STYLE to specify the style to be applied.

Example (using a Window style):

```
…
 OPEN WINDOW w1 WITH FORM “start_form"
   ATTRIBUTE (STYLE = “main")
…
```

In the .per file:

```
LAYOUT ( TEXT=“Customer Query”, STYLE=“newstyle” )
…
```

FOUR Js
The Power of Simplicity

DUI – Presentation Styles - Defaults

You can set the statusBarType attribute in the 4st style file for Windows to control the display of statusbar area.

Values of statusBarType can be:

none = No statusbar

default = The default status bar, same as panels2

panels1 = 1 panel, 1:comment/error/message

panels2 = 2 panels, 1:comment, 2:error/message

panels3 = 2 panels, 1:error/message, 2:comment

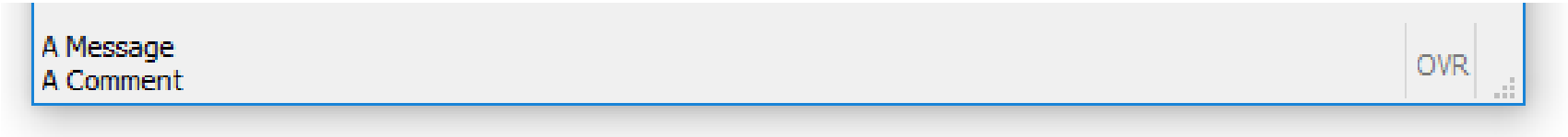
lines1 = 2 lines, 1:comment, 2:error/message

lines2 = 2 lines, 1:comment/message, 2:error

…

DUI – Presentation Styles - Example

```
<Style name="Window.statusbar">
  <StyleAttribute name="statusBarType" value="lines2" />
</Style>
```



DUI – Presentation Styles - Example

Call the following method in your program to use your style file instead of $FGLDIR/lib/default.4st:

```
CALL ui.Interface.loadStyles("mystylefile")
```

Style files have an extension .4st and are searched for in the FGLRESOURCEPATH

Use one of your styles in the application:

```
OPEN WINDOW w1 WITH FORM "order_form"
  ATTRIBUTE (STYLE = "mystyle")
```

Or the recommended way, on a form:

```
LAYOUT ( TEXT="Customer Query", STYLE="mystyle" )
```

FOUR Js
The Power of Simplicity

Do Chapter 3 - Exercise 1

The MENU statement

FOUR Js
The Power of Simplicity

DUI – Menu

Menu items are action views tied to the actions defined by the current interactive statement, which can be: DIALOG, MENU, INPUT, INPUT ARRAY, DISPLAY ARRAY or CONSTRUCT.

When a user selects a Menu item, the program receives an action event corresponding to the program trigger the item is bound to.

A Menu item is disabled automatically if that action is not available (for example, when a ring menu option is hidden).

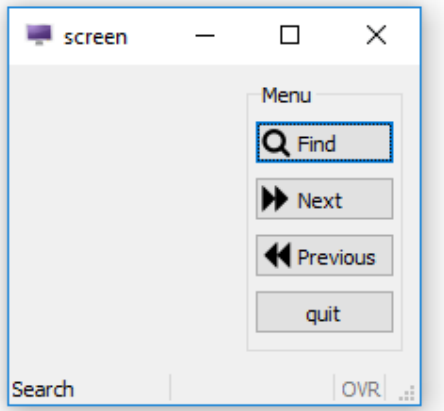
Since Menus are part of the Abstract User Interface, you can manipulate them dynamically at runtime.

DUI – Ring Menu

By default, Ring Menu buttons are generated in the application window for the actions listed in the MENU statement of your program code.

The menu buttons display in a specific area of the window, depending on the client software and current *style* applied (Action Panel).

```
1
2   MAIN
3     MENU "Menu"
4       ON ACTION find
5         CALL query_cust()
6       ON ACTION NEXT
7         CALL fetch_cust(1)
8       ON ACTION PREVIOUS
9         CALL fetch_cust(-1)
10      ON ACTION QUIT
11        EXIT MENU
12    END MENU
13  END MAIN
14
```

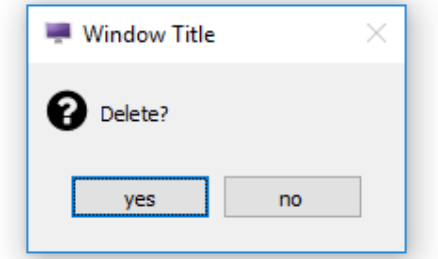


FOUR Js
The Power of Simplicity

DUI – Ring Menu

A Ring menu can also have styles applied to it to change how it's rendered, for example it can appear as a dialog window:

```
1  FUNCTION confirm_delete() RETURNS BOOLEAN
2    DEFINE l_ret BOOLEAN
3    MENU "Window Title"
4     ATTRIBUTE (STYLE="dialog", COMMENT="Delete?", IMAGE="question")
5      ON ACTION yes LET l_ret = TRUE
6      ON ACTION no LET l_ret = FALSE
7    END MENU
8    RETURN l_ret
9  END FUNCTION
10
11
12
```



DUI – Ring Menu

Some considerations:

By default the Ring Menu items appear on the ringMenu Panal. This panels position can be set in the styles file (.4st).

The ringMenu's panel position can be set to 'none' meaning it will not be visible.

An ON IDLE *idle-seconds* clause defines a set of instructions that must be executed after *idle-seconds* of inactivity.

Actions

DUI – Actions

The ON ACTION clause in your program executes a set of instructions.

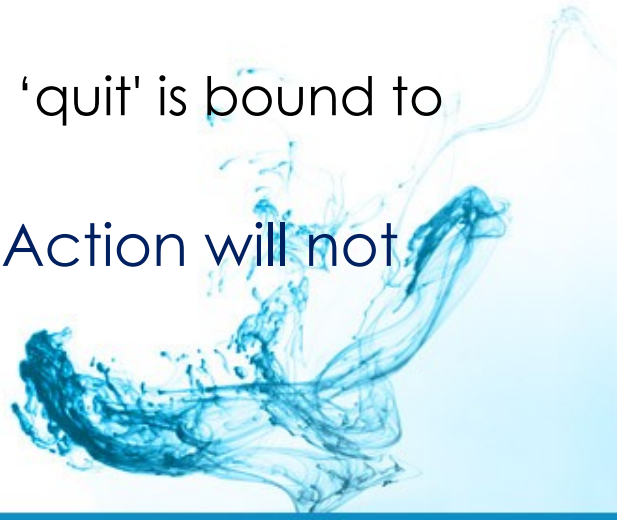
Can be used in DIALOG, MENU, INPUT, INPUT ARRAY, CONSTRUCT and DISPLAY ARRAY statements.

ON ACTION creates a default Action View in the application window, allowing the user to trigger the action.

You can define specific Action Views (like Topmenus or Toolbars in the form) and bind them to Actions using the "name" attribute.

> Example: a Toolbar item with the name 'quit' is bound to the Action having the name 'quit'.

If you define a specific Action View, that Action will not appear on the Action Panel.



FOUR Js
The Power of Simplicity

DUI – Actions - Syntax

```
-- In a dialog
INPUT ARRAY custarr WITHOUT DEFAULTS FROM sr_cust.*
  ON ACTION print
    CALL PrintRecord(custarr[arr_curr()].*)
  ON ACTION help
    CALL ShowHelp()
END INPUT
```

```
-- .per button definition (action view)
ATTRIBUTES
  BUTTON b1 : print, TEXT = "Print Record";
END
```

DUI – Actions

A list of Predefined Action Names is provided with the language.

For example:

INPUT / INPUT ARRAY / CONSTRUCT has predefined actions of ‘accept’ and ‘cancel’.

DISPLAY / INPUT ARRAY has predefined actions, ‘nextrow’ and ‘prevrow’, to bind action views for navigation.

The 'help' predefined action starts the help viewer for HELP clauses in dialog instructions.

```
INPUT BY NAME .... HELP 12423 <-- Creates action 'help'
```

DUI – Actions – Attribute Definitions

Define default attributes for graphical objects associated with actions (action views).

Centralize the decoration (appearance) of action views.

Can be defined in the form specification file (Action Defaults section) or in an external actions default file.

An external action defaults file is an XML file with the extension .4ad. Genero provides a default file, $FGLDIR/lib/default.4ad.

You can create your own action defaults file to replace default.4ad.

DUI – Actions – Attribute Definitions - Syntax

```
<ActionDefaultList>
  <ActionDefault name="action-name" [ attribute="value" [...] ] /> [...]
</ActionDefaultList>
```

Example:

```
<ActionDefaultList>
  <ActionDefault name="print" text="Print" image="printer“ comment="Print" />
  <ActionDefault name="modify" text="Update"comment="Update the record" />
</ActionDefaultList>
```

FOUR Js
The Power of Simplicity

DUI – Actions – Attributes

name: The name of action - Important: *This is case sensitve!*

acceleratorName: defines the primary accelerator key for an action.

acceleratorName2-4: additional accelerator keys for an action.

comment: defines tooltip hint for the user about the action.

contextMenu: defines whether a context menu option must be displayed for an action.

defaultView: default view (a button) must be displayed for a given action.

image: defines the image resource to be displayed for the action.

text: defines the label associated to the action.

validate: defines the data validation level for a given action.

Mobile Only:

disclosuseIndicator: drill-down decoration to an action.

rowbound: defines if the action is related to the row context of a record list.

DUI – Actions – Precedence

higher

lower

Attribute specified in the form for an element that is an action view (buttons, toolbar buttons, and topmenu options, for example)

Attribute specified for the action in the Action Defaults section of the form

Attribute specified for the action in a user-defined action defaults file (*filename*.4ad) or in the Genero-provided action defaults file (default.4ad)

DUI – Actions – Loading the .4ad

Specify a custom Action Defaults file in your program with the following Method:

```
CALL ui.Interface.loadActionDefaults("mydefaults")
```

Do Chapter 3 - Exercise 2

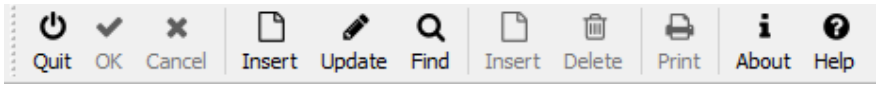
Toolbars

FOUR Js
The Power of Simplicity

DUI – Toolbars

A Toolbar is a view providing buttons that can trigger events in an interactive instruction.

The Toolbar items (buttons) are enabled in accordance with the actions defined by the current interactive instruction, which can be DIALOG, MENU, INPUT, INPUT ARRAY, DISPLAY ARRAY or CONSTRUCT.



DUI – Toolbars

A Toolbar can be defined in a form's TOOLBAR section; in this case the menu is available for that specific form only.

A TOOLBAR can also be:
Defined in a .4tb file.
Created from scratch in the 4gl with UI methods.

In these cases, the TOOLBAR is available to different forms.

FOUR Js
The Power of Simplicity

DUI – Toolbar – Definition

In the Form Specification File(.per):

```
TOOLBAR
  ITEM quit (TEXT="Quit",  IMAGE="quit")
  SEPARATOR
  ITEM find
  ITEM previous
  ITEM next
  SEPARATOR
  ITEM help
END
```

In an XML file (standard.4tb):

```
<ToolBar>
  <ToolBarItem name="quit" text="Quit" image="quit" />
  <ToolBarSeparator/>
  <ToolBarItem name="query" text="Query" image="search" />
  <ToolBarItem name="add" text="Append" image="add" />
  <ToolBarItem name="delete" text="Delete" image="delete" />
  <ToolBarItem name="modify" text="Modify" image="change" />
  <ToolBarSeparator/>
  <ToolBarItem name="f1" text="Help" image="list" />
</ToolBar>
```

DUI – Toolbar – Definition

Node	Attribute	Type	Description
Toolbar	buttonTextHidden	INTEGER	Defines if the texts of toolbar buttons must appear by default
ToolbarItem	Name	STRING	Defines the action corresponding to the toolbar button.
ToolbarItem	Text	STRING	Defines the text to be displayed in the toolbar button.
ToolbarItem	Comment	STRING	Defines the message to be shown as tooltip when the user selects a toolbar button.
ToolbarItem	Image	STRING	Defines the icon to be used in the toolbar button.
ToolbarItem	hidden	INTEGER	Indicates if the item is hidden
ToolbarSeparator	hidden	INTEGER	Indicates if the separator is hidden

DUI – Toolbar – Loading

You can define a Toolbar in an XML file and load it at runtime using the utility method provided by the Interface built-in class:

```
CALL ui.Interface.loadToolbar("standard")
```

Or you can use the utility method provided by the Form built-in class:

```
DEFINE myform ui.form
…
CALL myform.loadToolbar("standard")
```

NOTE: This method does require you to get the form object first!

DUI – Toolbar – Summary

Some considerations:

The DOM tag names are case sensitive, *Toolbar* is different from *ToolBar*.

When binding to an action, make sure that you are using the right value in the **name** attribute.

When binding to a key, make sure the **name** attribute value is in lowercase eg: "f5".

Make sure the image specified is accessible to the Genero Client.

Do Chapter 3 - Exercise 3

FOUR Js
The Power of Simplicity

TopMenus

FOUR Js
The Power of Simplicity

DUI – TopMenu

A Top Menu is a view for actions presented as a typical pull-down menu that can trigger events.

Menu options are bound to the actions specified in the program by ON ACTION clauses.

The Top Menu items (buttons) are enabled in accordance with the actions defined by the current interactive statement in your program, which can be DIALOG, MENU, INPUT, INPUT ARRAY, DISPLAY ARRAY or CONSTRUCT.

DUI – TopMenu - Defining

A Top Menu can be defined in a form's TOPMENU section; in this case the menu will be available for the specific form only.

Defined in a .4tm file and loaded using a **ui.Form** object method of **loadTopMenu()** or loaded at a user interface level using **ui.Interface.loadTopMenu()**

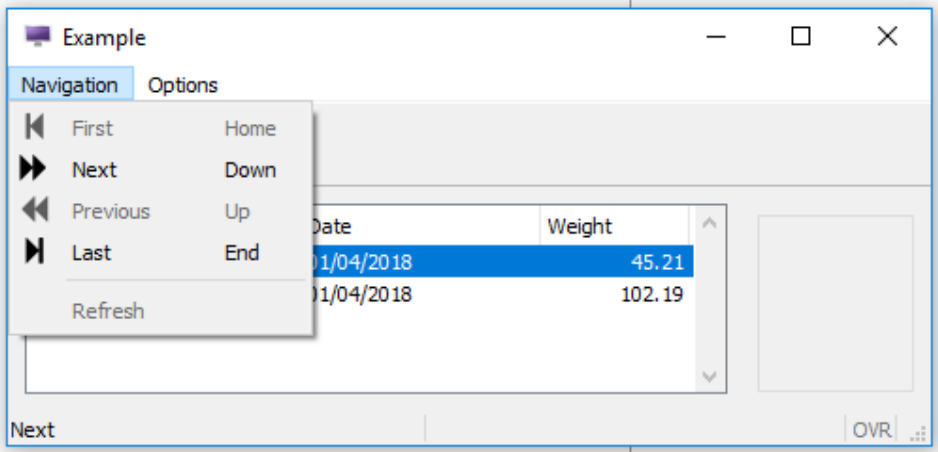
Created from scratch in the 4gl with UI methods.

Those last two method mean the TopMenu is re-usable in other forms.

DUI – TopMenu - .per Syntax

```
TOPMENU
group
   [...]
[END]
where group is:
 GROUP group-identifier ( group-attribute [,...] )
   { command
   | group
   | SEPARATOR
   } [...]
END
where command is:
 COMMAND command-identifier ( item-attribute [,...] )
```

```
TOPMENU
  GROUP navi (text="Navigation")
    COMMAND firstrow (text="First")
    COMMAND nextrow (text="Next")
    COMMAND prevrow (text="Previous")
    COMMAND lastrow (text="Last")
    SEPARATOR
    COMMAND refresh (text="Refresh")
  END -- GROUP
  GROUP options (text="Options")
    COMMAND accept
    (text="Select",image="ok")
    COMMAND cancel
    (text="Close",image="cancel")
  END -- GROUP
END -- TOPMENU
```



FOUR Js
The Power of Simplicity

DUI – TopMenu - .4tm - XML Syntax

```
<TopMenu [ topmenu-attribute="value" [...] ] >
  group
  [...]
</TopMenu>
```

Where *group* is:

```
<TopMenuGroup group-attribute="value" [...]>
  { <TopMenuSeparator/>
  | <TopMenuCommand command-attribute="value" [...] />
  | group
  } [...]
</TopMenuGroup>
```

Example:

```
<TopMenu>
    <TopMenuGroup text="File" >
        <TopMenuCommand name="clear" text="Clear" image="new" />
        <TopMenuSeparator/>
        <TopMenuCommand name="quit" text="Quit“ />
    </TopMenuGroup>
    <TopMenuGroup text="Edit" >
        <TopMenuCommand name="editcut" />
        <TopMenuCommand name="editcopy" />
        <TopMenuCommand name="editpaste” />
    </TopMenuGroup>
</TopMenu>
```

DUI – TopMenu - .4tm - XML Syntax

Node	Attribute	Type	Description
TopMenuGroup	text	STRING	Defines the text to be displayed in the pull-down menu group
TopMenuGroup	comment	STRING	Defines the message to be shown for this element
TopMenuGroup	image	STRING	Defines the icon to be used in the pull-down menu group
TopMenuGroup	hidden	INTEGER	Indicates if the group is hidden

DUI – TopMenu - .4tm - XML Syntax

Node	Attribute	Type	Description
TopMenuCommand	name	STRING	*Defines the action corresponding to the topmenu command*
TopMenuCommand	text	STRING	Defines the text to be displayed in the pull-down menu option
TopMenuCommand	comment	STRING	Defines the message to be shown for this element
TopMenuCommand	image	STRING	Defines the icon to be used in the pull-down menu option
TopMenuCommand	hidden	INTEGER	Indicates if the command is hidden
TopMenuSeparator	hidden	INTEGER	Indicates if the separator is hidden

DUI – TopMenu – Loading

If you defined a TopMenu in an XML file you can load it at runtime using the utility method provided by the Interface built-in class:

```
CALL ui.Interface.loadTopMenu("standard")
```

Or you can use the utility method provided by the Form built-in class:

```
CALL ui.Window.getForm().loadToolbar("standard")
```

DUI – TopMenu - Summary

You can define an XML file to create your TopMenu.

The DOM tag names are case sensitive; *Topmenu* is different from **TopMenu**.

A TopMenu is part of a form definition (the TopMenu node must be created under the Form node).

A TopMenu command is bound to an *action node* of the current interactive instruction if its name attribute corresponds to an *action node* name.

Do Chapter 3 - Exercise 4

FOUR Js
The Power of Simplicity

StartMenu

FOUR Js
The Power of Simplicity

DUI – StartMenu

A Start Menu defines a tree of commands that start programs on the application server where the runtime system executes.

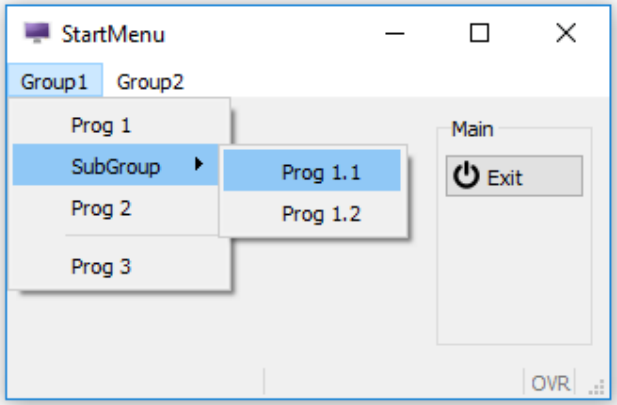
We recommend that you create a specific BDL program dedicated for running the Start Menu. This program must create (or load) a Start Menu and then perform an interactive instruction (eg. DISPLAY ARRAY or a MENU in the SCREEN window)

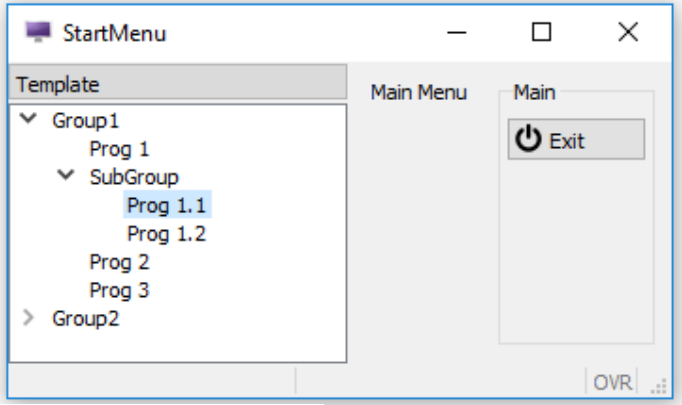
A program can only have one Start Menu.

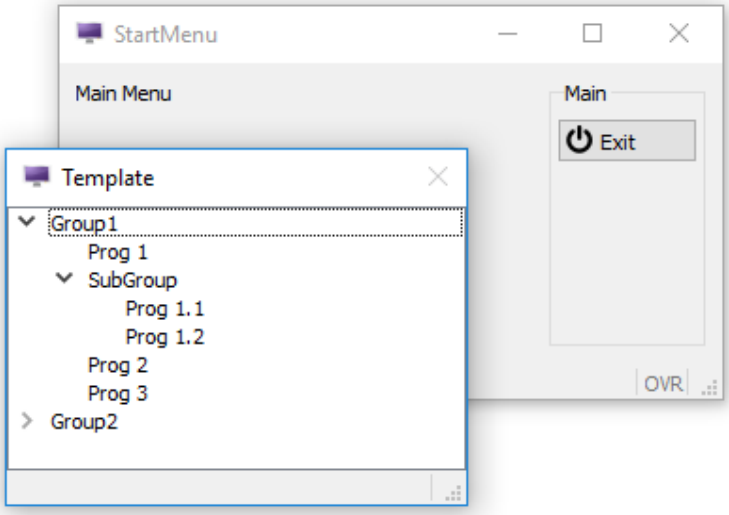
The Start Menu window should use the style 'main'.

DUI – StartMenu

A StartMenu can be one of three styles: Menu, Tree or Poptree.







FOUR Js
The Power of Simplicity

DUI – StartMenu – XML Syntax

```
<StartMenu [ startmenu-attribute="value" [...] ] >
  group
  [...]
</StartMenu>
```

where startmenu-group is:

```
<StartMenuGroup group-attribute="value" [...]>
  { <StartMenuSeparator/>
  | <StartMenuCommand command-attribute="value"
[...] />
  | group
  } [...]
</StartMenuGroup>
```

```
<?xml version="1.0" encoding="ANSI_X3.4-1968"?>
<StartMenu text="Template">
  <StartMenuGroup text="Group1">
    <StartMenuCommand text="Prog 1" exec="fglrun prog1.42r"/>
    <StartMenuGroup text="SubGroup">
      <StartMenuCommand text="Prog 1.1" exec="fglrun prog1.42r Command 1.1"/>
      <StartMenuCommand text="Prog 1.2" exec="fglrun prog1.42r Command 1.2"/>
    </StartMenuGroup>
    <StartMenuCommand text="Prog 2" exec="fglrun prog2.42r Command 2"/>
    <StartMenuSeparator/>
    <StartMenuCommand text="Prog 3" exec="fglrun prog3.42r"/>
  </StartMenuGroup>
  <StartMenuGroup text="Group2">
    <StartMenuCommand text="Demo" exec="fglrun demo.42r"/>
  </StartMenuGroup>
</StartMenu>
```

DUI – StartMenu – XML Syntax

Node	Attribute	Type	Description
StartMenu	text	STRING	Defines the text to be displayed as title.
StartMenuGroup	text	STRING	Defines the text to be displayed for this group.
StartMenuGroup	image	STRING	Defines the icon to be used for this group.
StartMenuGroup	disabled	INTEGER	Indicates if the group is disabled (grayed out, cannot be selected.)
StartMenuCommand	**exec**	STRING	Defines the command to be executed when the user selects this command.
StartMenuCommand	text	STRING	Defines the text to be displayed for this command.
StartMenuCommand	Image	STRING	Defines the icon to be used for this command.
StartMenuCommand	disabled	INTEGER	Indicates if the group is disabled (grayed out, cannot be selected.)

DUI – StartMenu – Loading

You can use the utility method provided by the Interface built-in class.

```
CALL ui.Interface.loadStartMenu("app_menu")
```

"app_menu" is a file with an extension .4sm located in the FGLRESOURCEPATH.

DUI – StartMenu – Summary

You can define an XML file to create your start Menu.

The DOM tag names are case sensitive; *Startmenu* is different from **StartMenu**.

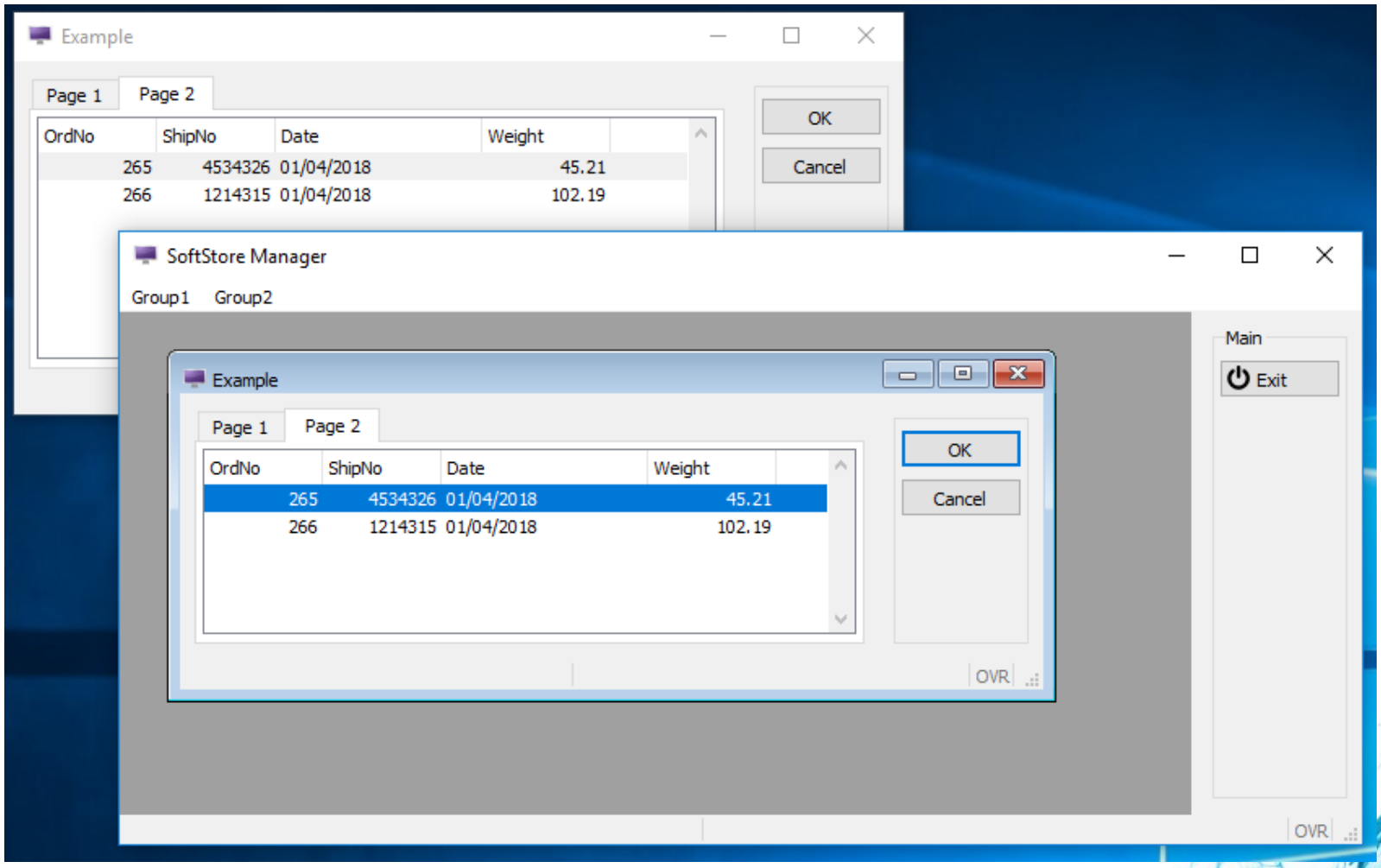
You can apply different decorations for a StartMenu : Menu | tree | poptree (see Window Style attributes).

Since the Start Menu is part of the Abstract User Interface, you can manipulate it at runtime.

Do Chapter 3 - Exercise 5

MDI

DUI – SDI vs MDI



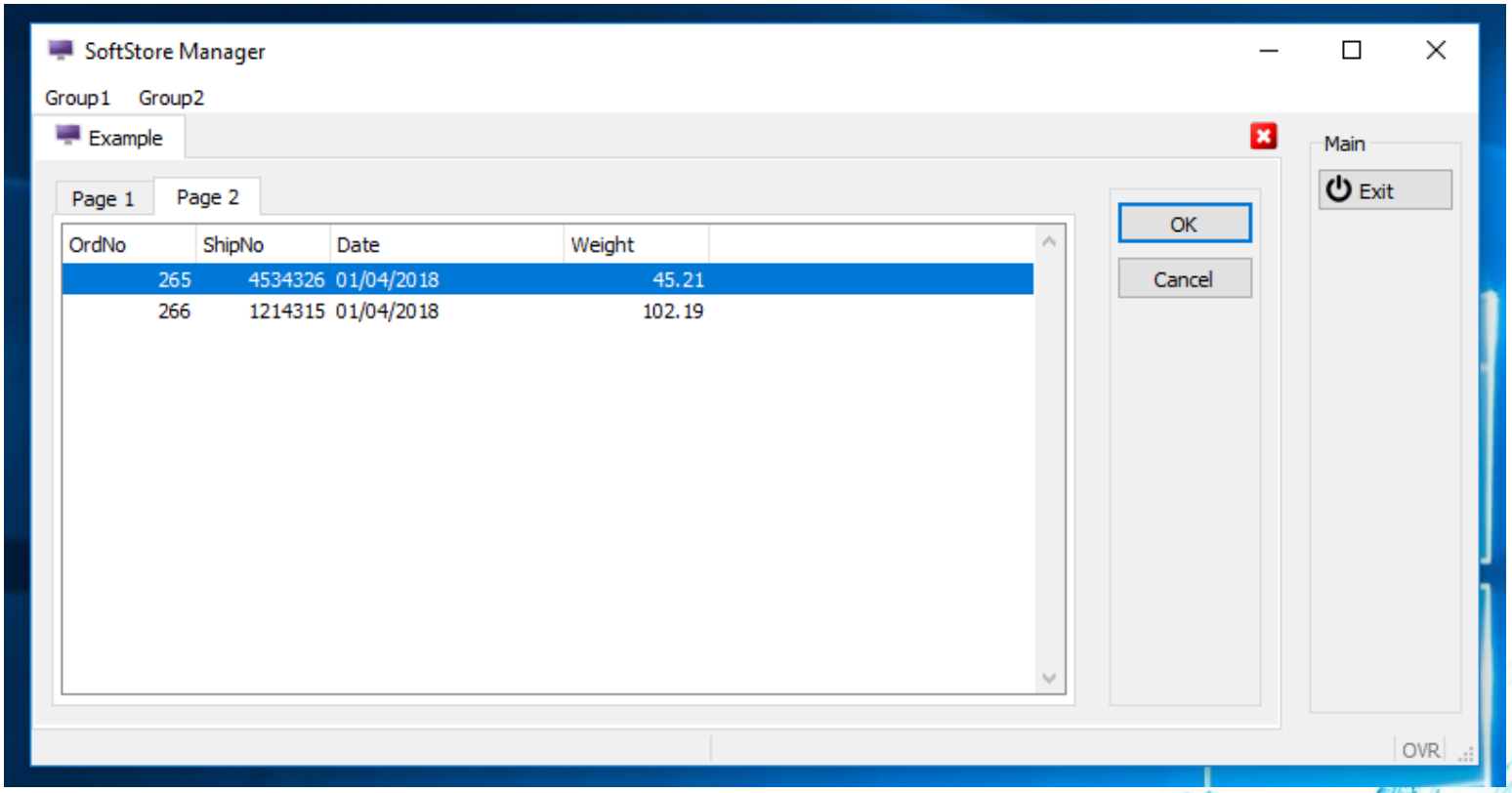
FOUR Js
The Power of Simplicity

DUI – MDI – Tabbed Container

```
<Style name="Window.main">
   <StyleAttribute name="windowType" value="normal" />
   <StyleAttribute name="startMenuPosition" value="menu" />

   <StyleAttribute name="tabbedContainer" value="yes" />

</Style>
```



FOUR Js
The Power of Simplicity

DUI – MDI

In BDL, Multiple Document Interface is also referred to as WCI: Window Container Interface.

WCI makes sense when the application programs have a main window that is the root for other windows opened temporarily (typically, modal windows to pick a record in a list).

WCI configuration is done dynamically at the beginning of programs, by using some methods of the ui.Interface built-in class.

The WCI container program is a separate BDL program of a special type, in which the default SCREEN window is used as an MDI container.

DUI – MDI

The WCI container program must indicate that its type is special (setType), and must identify itself (setName):

```
MAIN
  CALL ui.Interface.setName("parent1")
  CALL ui.Interface.setType("container")
  CALL ui.Interface.setText("SoftStore Manager")
  CALL ui.Interface.loadStartMenu("mystartmenu")
  …
```

WCI children programs must attach to a parent container by giving the name of the container program:

```
MAIN
  CALL ui.Interface.setName("custapp")
  CALL ui.Interface.setType("child")
  CALL ui.Interface.setContainer("parent1")
…
```

DUI – MDI - Summary

The only style applied to the container is Window.main.

The Toolbar is unique for the container, however the StartMenu and TopMenu are not.

If a child program is started, but the parent container is not, the client shows an error message and the program stops with a runtime error.

If a container program is run twice, the client shows an error message and the program stops with a runtime error.

When the parent container program is stopped, it's child applications are terminated.

DUI – Summary

All these objects can be created and manipulated in Genero:

- Windows and Forms
- Presentation Styles
- Statusbar
- Ring Menus
- Actions
- Toolbars
- Top Menus
- Start Menus
- Multiple Document Interface (MDI/WCI)

FOUR Js
The Power of Simplicity

Q&A

Intelligent Business Application Infrastructure