GUI Design Studio

Quick Start Tutorial Manual

Version 2.4
October 2007

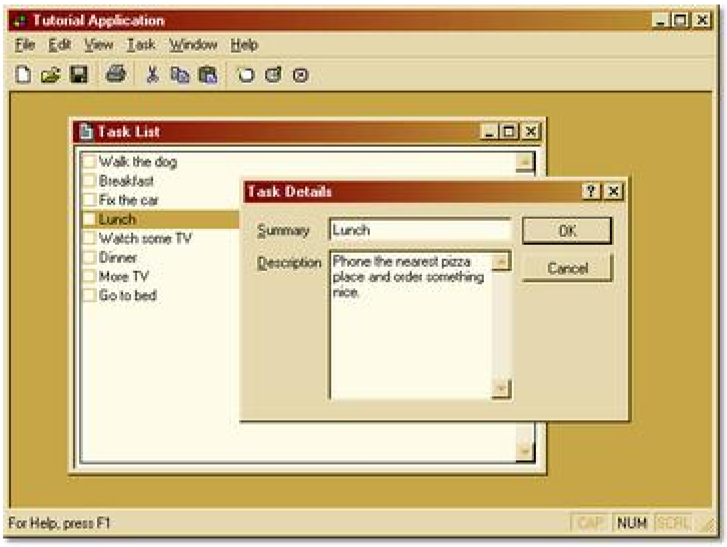
Contents

Quick Start Tutorial

This tutorial provides a brief introduction to the GUI Design Studio product and its main features. You may be able to learn enough simply by reading it through but the best way to learn is to follow the steps. It should take between 5 and 10 minutes to complete each of the 11 steps.

All screenshots of design windows in this tutorial have been taken using the "Windows Classic" look and feel and the "Maple" design colour scheme as set in the Preferences ("File | Preferences..." menu command). This helps to distinguish them from application dialogs and windows that appear in the standard Windows XP colour scheme. If you work with a different look and feel or colour scheme then your designs will obviously look a bit different.

The tutorial uses the hypothetical "Task List Manager" application shown below:



Summary: Lunch

Description: Phone the nearest pizza place and order something nice.

This example is not intended to represent a good application or necessarily a good user interface for such an application. It does, however, provide a simple introduction to most of the basic building blocks that you can use in your own application designs.

The tutorial is in two parts. The first deals with the essential topics of creating projects, design documents and basic design procedures. The second part introduces storyboarding which allows you to create interactive prototypes to demonstrate your ideas.

- Tutorial project files
- Part 1: Designing the components
- Part 2: Bringing the design to life

Tutorial project files

All of the project files created in this tutorial have been supplied as samples. The name of the related file (that you will duplicate by following the tutorial) is given at the start of each step where applicable. The files are also listed below with their associated steps:

Part 1: Designing the components

Tutorial.gdp	Step 1: Creating the project
NewTaskDialog.gui	Step 2: Creating the New Task dialog
ApplicationWindow.gui	Step 3: Creating the Application Window
TaskListDocument.gui	Step 4: Creating a Document Window
EditTaskDialog.gui	Step 5: Creating the Edit Task dialog
Screenshot.gui	Step 6: Putting together a screenshot

Part 2: Bringing the design to life

Storyboard1.gui	Step 1: Creating a simple storyboard
Storyboard2.gui	Step 2: Adding navigational features
EmptyDocument.gui	Step 3: Adding a standard File Open dialog
Storyboard3.gui	Step 3: Adding a standard File Open dialog
Storyboard4.gui	Step 4: Adding a menu system

Part 1: Designing the components

Although it is quite possible to create one big design document that contains all of the windows and dialogs that make up an application, it is better to separate them out into their component parts. This makes the overall design much more manageable.

The Tutorial Application has three main components:

1. The main application window.
2. A task list document window (2 variants).
3. A task details dialog (2 variants).

Part 1 covers the essential steps required to create a design project and build some basic user interface components.

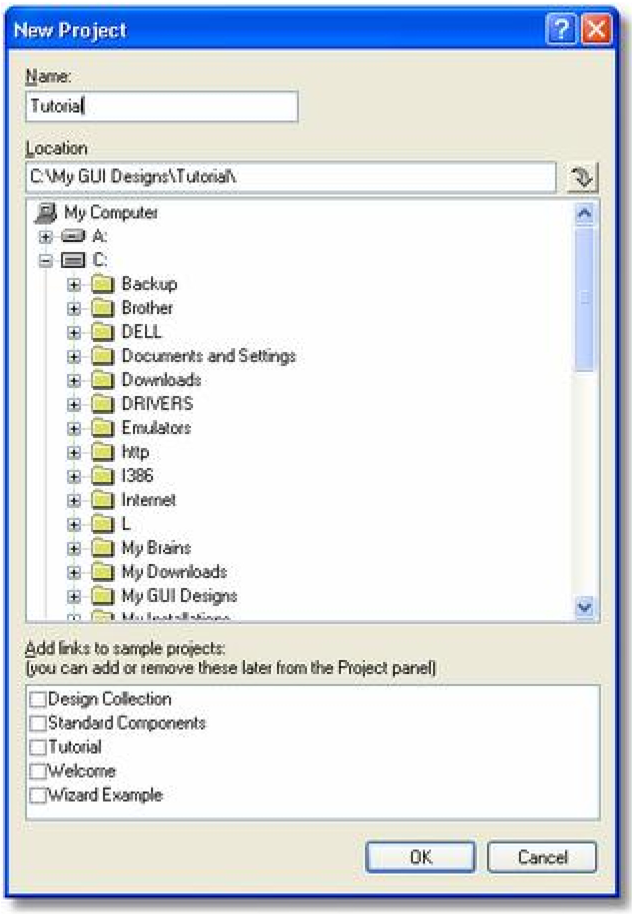
- Step 1: Creating the project
- Step 2: Creating the New Task dialog
- Step 3: Creating the Application Window
- Step 4: Creating a document Window
- Step 5: Creating the Edit Task dialog
- Step 6: Putting together a screenshot
- Step 7: Previewing the result
- Part 1 Summary

Step 1: Creating the project

Tutorial Project File: Tutorial.gdp

GUI Design Studio allows you to create individual design documents at will. This is useful if you want to create a quick design of something. However, most of your work will involve multiple design documents organised into projects.

1. Use the "File | New Project..." menu command to open the "New Project" dialog:



New Project

Name: Tutorial

Location: C:\My GUI Designs\Tutorial\

Add links to sample projects:
(you can add or remove these later from the Project panel)

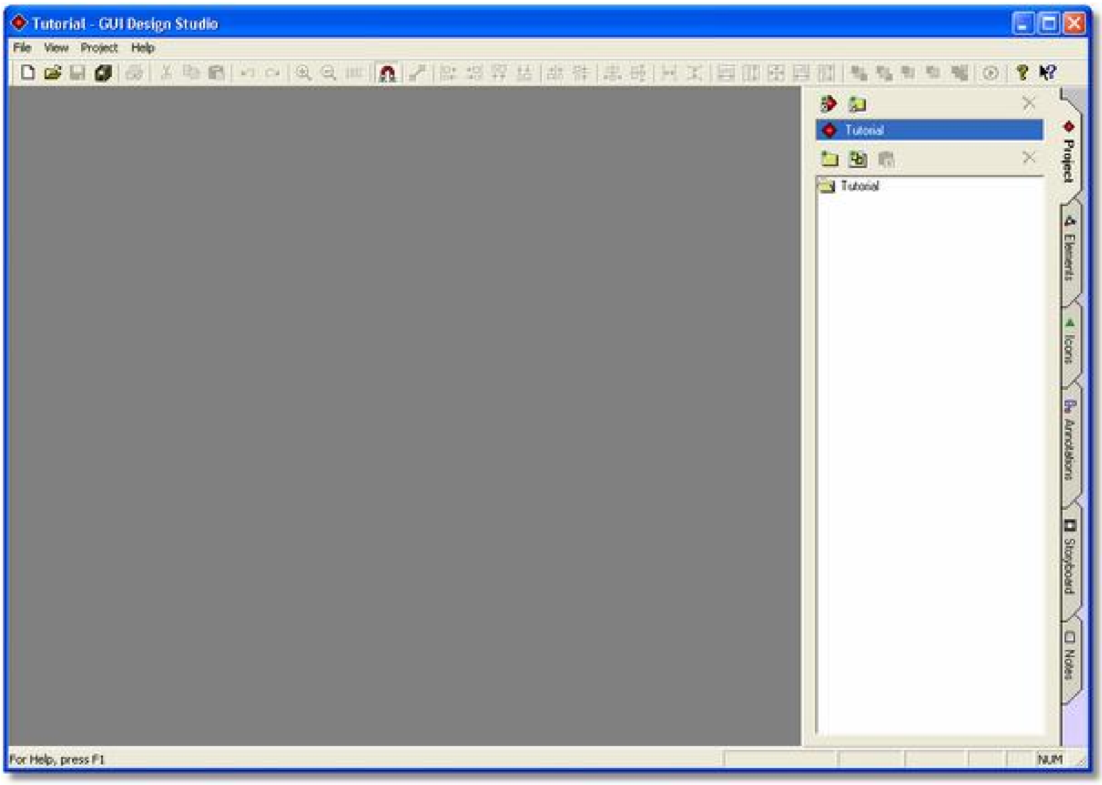
☐ Design Collection
☐ Standard Components
☐ Tutorial
☐ Welcome
☐ Wizard Example

2. Select a suitable folder in which to create the project such as "C:\My GUI Designs\" shown above. If the folder you want to use does not already exist, just type it into the Location box and it will be created for you with the new project.

3. Type "Tutorial" into the Name box. This will automatically create a "Tutorial" sub-folder for your project.

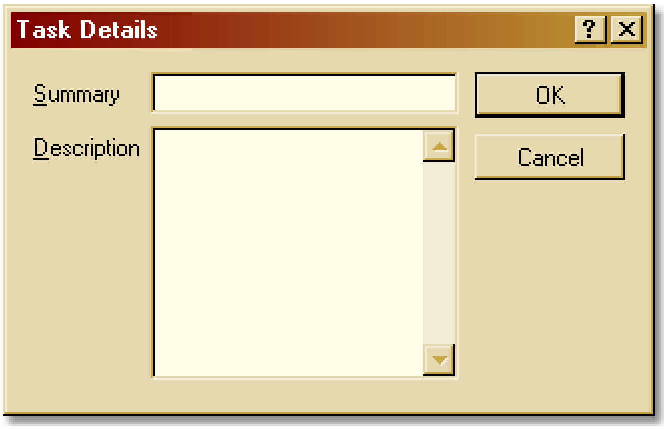
4. For the purposes of this tutorial, leave all of the sample project links unchecked. When you create your own projects, you may wish to check the "Standard Components" or "Design Collection" projects to include links to them in your project but you can always do this later as you will see in Part 2 of this tutorial.

5. Hit "OK" and the new project will be created and opened. Your screen should now look something like this (if your Design Bar is on the left you can move it to the right using the "View | Design Bar Position | Dock Right" menu command if you wish):


Tutorial - GUI Design Studio
File
View
Project
Help
Tutorial
Tutorial
Project
Elements
Icons
Annotations
Storyboard
Notes
For Help, press F1
NUM

Step 2: Creating the New Task dialog

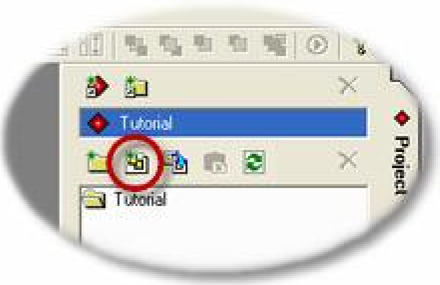
Tutorial Project File: NewTaskDialog.gui



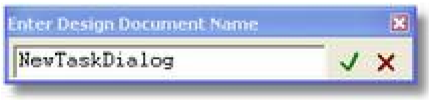
The first component we shall create is the dialog that is to appear when the user wants to create a new task. This is a very simple design to get you started.

1. Although you can create a new design document using the "File | New" menu command (Ctrl+N), this will produce an untitled document that will need to be saved later in a suitable location and with a suitable name. Since you normally know what you are going to create and where you want it, GUI Design Studio allows you to create new files directly into the desired project folder.

   With the "Project" Design Panel open, click on the "New Design" command icon:



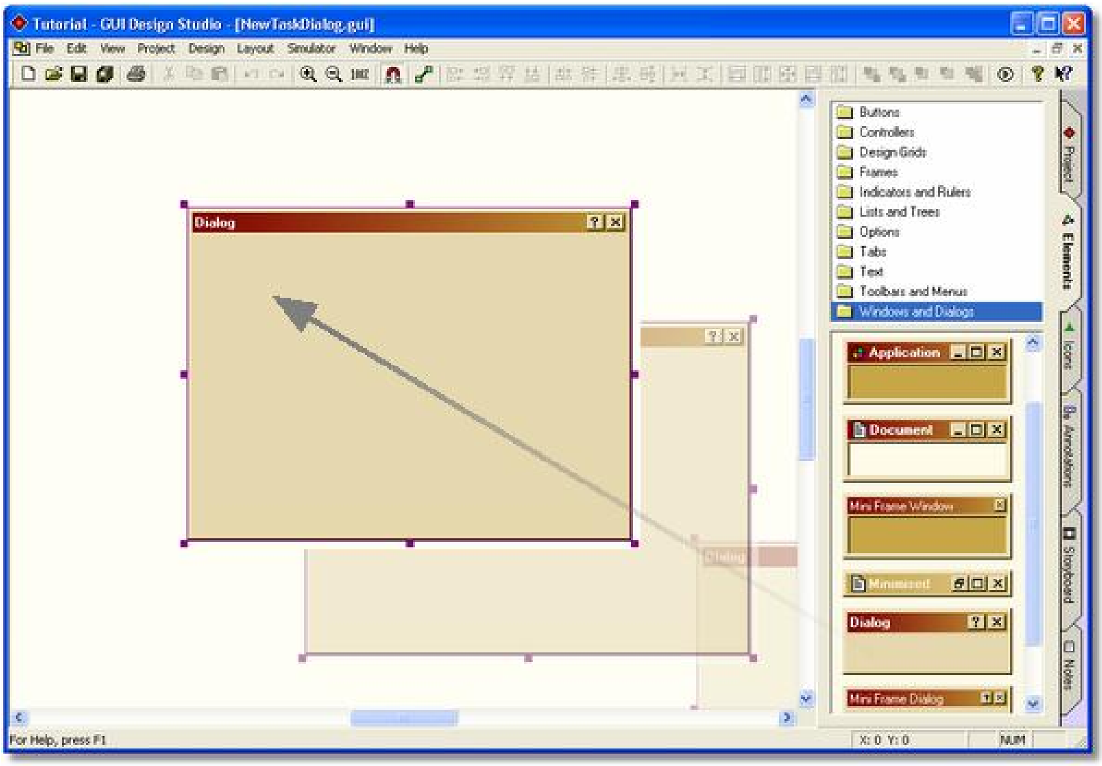
2. Enter the name "NewTaskDialog" when prompted:



NewTaskDialog

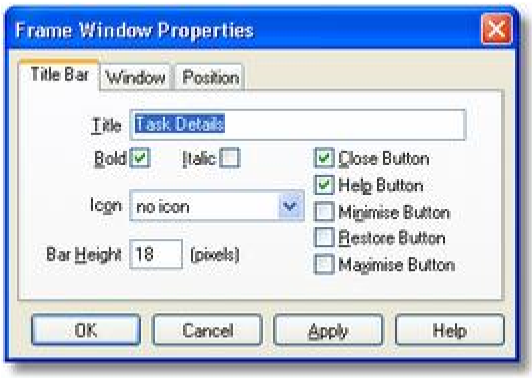
3. Select the "Elements" Design Panel and the "Windows and Dialogs" category.

4. Locate the "Dialog" element in the list then drag and drop it onto your design:



5. Use the resize handles to make the dialog a bit smaller.

6. Double-click the dialog to open its property editor and change the Title to "Task Details". Note that the dialog is actually a type of Frame Window. Hit "OK" to accept the change.


Frame Window Properties

Title Bar | Window | Position

Title: Task Details

Bold ☑ Italic □

Icon: no icon

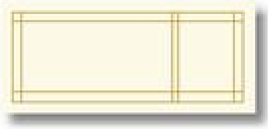
Bar Height: 18 (pixels)

☑ Close Button
☑ Help Button
□ Minimise Button
□ Restore Button
□ Maximise Button

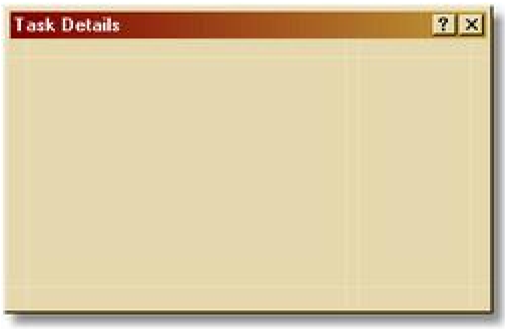
7. Ensure that the "Layout | Snap to Edges" menu command is checked. Edge snapping is a very powerful feature of GUI Design Studio and you will normally want to work with this turned on. Unlike traditional grid snapping, edge snapping allows GUI elements to correctly align with each other rather than an arbitrary background grid.

8. With dialogs, it is useful to have consistent margins and placement of exit buttons. This is not essential when putting together a quick design but it does add a professional touch. To achieve this, we will use a design grid.

   Select the "Design Grids" category and locate the following grid with margins all around and a space for buttons down the right-hand side:

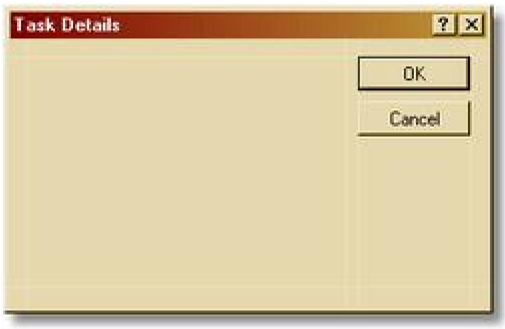


Now drag and drop this on top of the dialog so that it 'snaps' into the top-left corner of the dialog. Use the bottom-right drag handle to snap the other corner into the dialog. You should now have something that looks a bit like this:



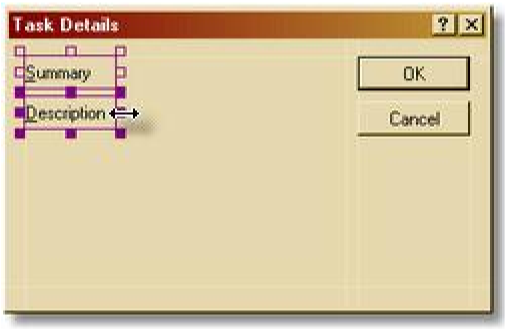
9. The "Buttons" category contains a selection of common buttons for convenience but you can change them once you've added them to your design.

   Use drag and drop to add "OK" and "Cancel" buttons, snapping them onto the grid. You will note that the default size of the buttons and the right-hand column of the grid are the same.



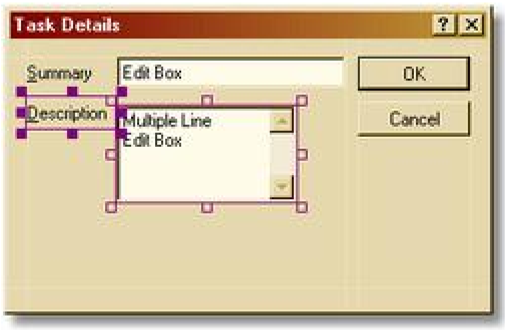
10. Switch to the "Text" category and add a couple of "Left Text" elements to use as prompts, snapping them to the left-hand side of the grid. Double-click them to edit their properties and change the Text to "&Summary" and "&Description". The ampersand (&) underlines the next character to indicate a keyboard accelerator.

11. Click on the "Summary" prompt to select it then Ctrl+Click on the "Description" prompt so that they are both selected. You can now resize them together so that they are the right width for the longest prompt:

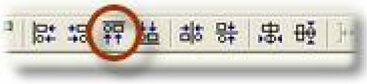


12. The edit boxes should now be fairly easy. The first, single line edit box will snap nicely up to the "Summary" prompt and can then be extended to the right-hand side of the grid.

13. The multiple line edit box is a bit more tricky because there is only one edge to snap to. Place it next to the "Description" prompt and then Ctrl+Click the prompt so that they are both selected:

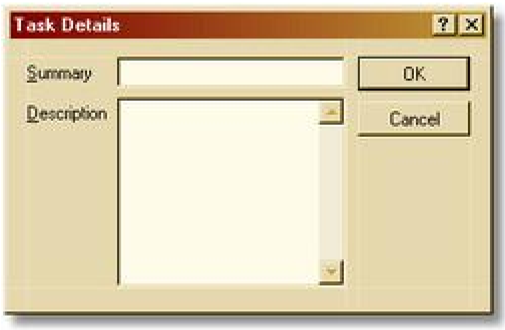


14. Now use the "Layout | Align | Top" menu command or the following toolbar button to align the top of the edit box to that of the prompt:



Now that the top-left is aligned correctly, the bottom-right can be dragged out to snap into the corner of the grid cell.

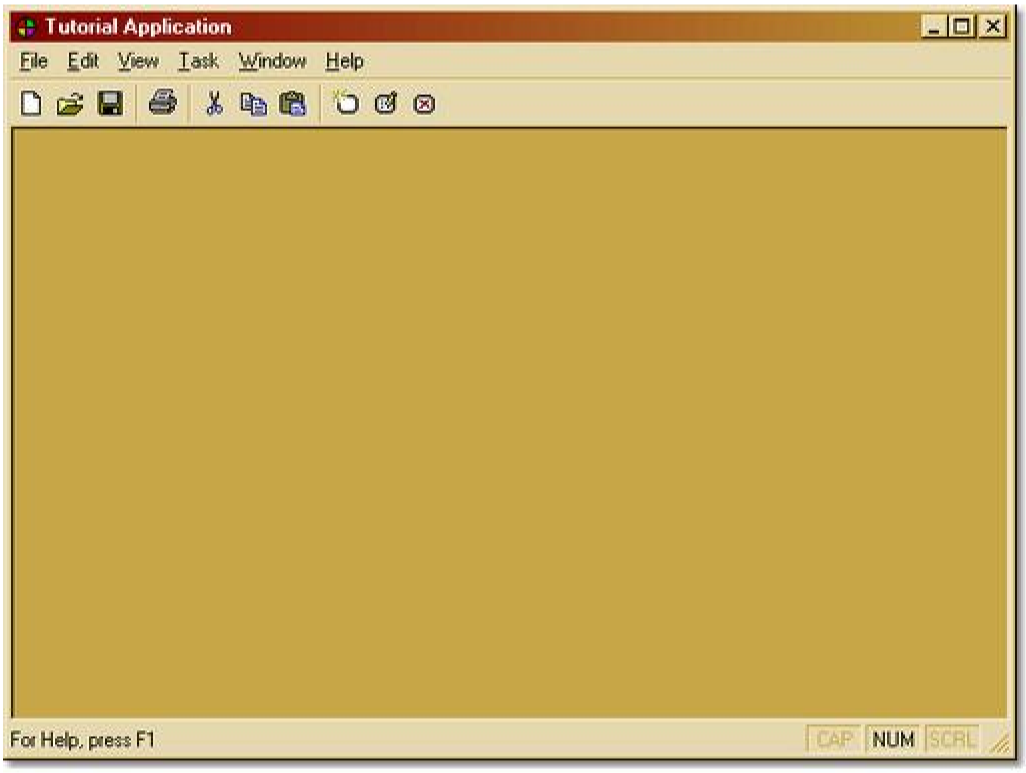
15. Finally, change both the edit box properties to set their Text fields to blank and save your work. You should end up with a dialog looking something like this:



It probably took longer to explain and read than it actually took to complete this step. With practice, it can be done in just a couple of minutes. You should now know enough to be able to create more complex dialogs. Explore the different alignment, sizing and spacing tools to see how they can help you.

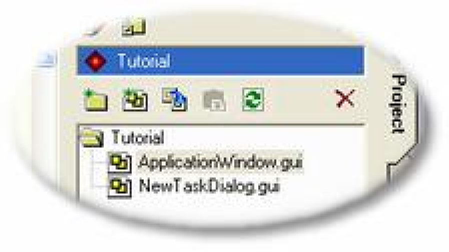
Step 3: Creating the Application Window

Tutorial Project File: ApplicationWindow.gui



This is a typical multiple document interface (MDI) application window with a menu bar, toolbar and status bar.

1. Start by creating a new design document called "ApplicationWindow" in the project folder in the same way that you created the "NewTaskDialog" in Step 2. The document will be opened ready for creating the design and your Project tree should now look like this:

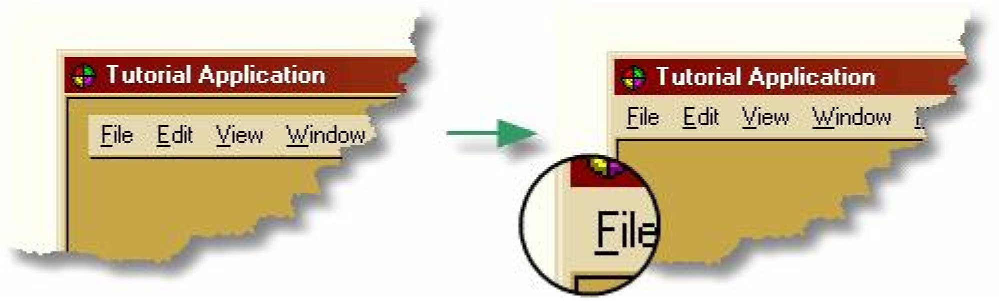


2. From the "Elements" panel, locate the Application "Frame Window" element in the "Windows and Dialogs" category and add this to the design.

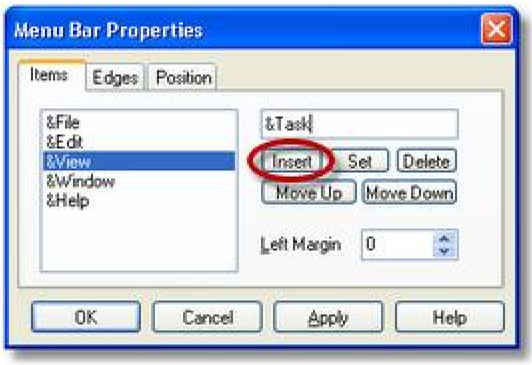
3. Resize the window to make it larger then edit the properties (double-click or press Alt+Enter) and change the title to "Tutorial Application".

4. Now add a "Menu Bar" from the "Toolbars and Menus" category. You should be able to snap it right into place under the title bar of the window. You can then extend the width to snap in the other end.

   Notice that the menu edges overlap the sunken border of the frame window to create a new border and neat corners:



5. We are now going to insert a new "Task" menu item between "View" and "Window". Open the menu properties editor and select "&View" in the list of items. Now type "&Task" into the edit box, hit the "Insert" button (Alt+I shortcut key) then "OK" to accept the change:



Menu Bar Properties

Items: &File, &Edit, &View, &Window, &Help

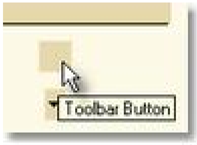
&Task

Left Margin: 0

6. Next add a "Docked Toolbar" under the menu just as you did for the Menu Bar. Notice that the top edge is grooved to make a suitable join with the menu bar and the bottom has a replacement sunken edge. If you ever want a toolbar without a menu, you can make the top edge flat in the properties.

7. Now that we have a blank toolbar, we can add buttons (and other controls if desired). Buttons do not have image properties. Instead these must be added afterward.

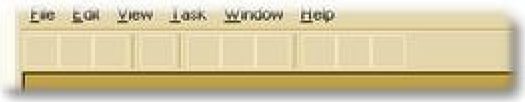
   By default, the "Toolbar Button" element has the 'flat' style which makes it less than obvious in the palette but the tool tip soon gives it away:



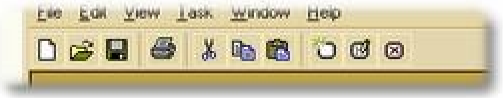
   When you add a 'flat' toolbar button (or a non-flat spacer) to a toolbar, a light border is shown. Without this, it would be impossible to see! The border does not appear when the design is printed or exported.

   Referring to the image at the start of this step, you can see that we want to add 10 buttons with 3 spacers to group them. The buttons and spacers are designed to butt up to each other and like all other elements, they snap together. The toolbar itself has hidden rails to make it easy to place them using the default sizes.

   You should end up with something like this:



8. Now that we have buttons, we can add the icons. Switch to the "Icons" Design Panel and the "Toolbars" category. Locate suitable command icons and place them on top of the buttons. They should snap into place:



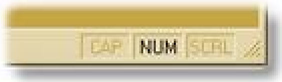
   The first 7 icons are for the familiar New, Open, Save, Print, Cut, Copy and Paste commands. The last 3 are intended to represent commands for creating, editing and deleting Tasks in our application.

9. With the toolbar complete, the last part of this step is to create the "Status Bar". This can be found in the "Toolbars and Menus" category back on the "Elements" panel. Snap the status bar into place at the bottom of the application window and resize it to full width.

10. For maximum design flexibility, the status bar does not contain any indicators. Just like the toolbar with its buttons, the indicators must be added manually. Before that however, we shall add a corner resize gripper in the bottom-right corner. Because it is often used with scroll bars on document windows, this can be found in the "Controllers" category:

11. Finally, add the "Status Indicator" elements to the status bar from the "Toolbars and Menus" category. These are designed to butt up together with a natural visual gap between them and sit on hidden rails on the status bar.

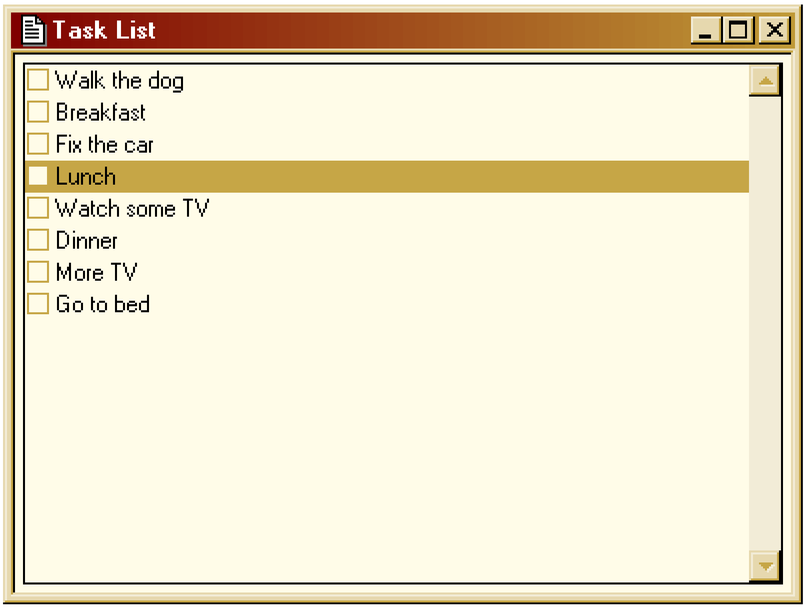
    Add 3 indicators and edit their properties to create standard "CAP", "NUM" and "SCRL" indicators for Caps Lock, Num Lock and Scroll Lock respectively. Make just the "NUM" indicator active:



12. You have now completed the design of the application window so save your work.

Step 4: Creating a Document Window

Tutorial Project File: TaskListDocument.gui



Task List

□ Walk the dog
□ Breakfast
□ Fix the car
□ Lunch
□ Watch some TV
□ Dinner
□ More TV
□ Go to bed

Having completed steps 2 and 3, you should now be able to create almost any design you like. The purpose of this step is to create another component for our Task List Manager application and to introduce annotations. The design has therefore been kept to a minimum and consists just 2 elements!

1. As before, switch to the "Project" panel and create a new design document. Call it "TaskListDocument".

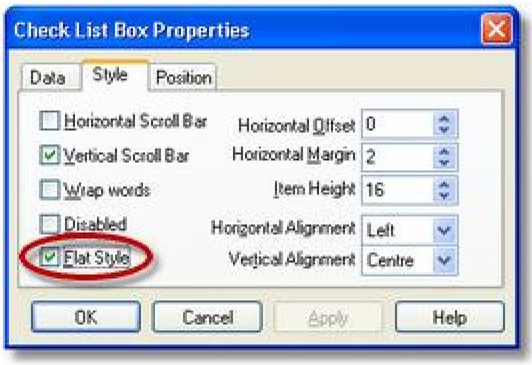
2. From the "Elements" panel, locate the Document "Frame Window" element in the "Windows and Dialogs" category and add this to the design.

3. You can keep the window at its default size but edit its properties to change the title to "Task List".

4. From the "Lists and Trees" category, place a "Check List Box" element on top of the document window. Because of the border on the list box, it looks a little better to leave a margin around the edge. This could have been done with a Design Grid like the one used in Step 2: Creating the New Task dialog, but in this instance it is probably easier just to resize it manually.

   One good way is to snap it right up to the edges of the document frame and then use the cursor keys to accurately move and resize it. Use the Right and Down arrow keys to offset the top-left by 4 pixels each then use Shift+Left and Shift+Up to shrink the list box by 8 pixels each. This will leave a neat 4-pixel margin all around.

5. Open the property editor for the list box (double-click or Alt+Enter with the list box selected). Switch to the "Style" tab first and check the "Flat Style" option to give the list box a thin border:



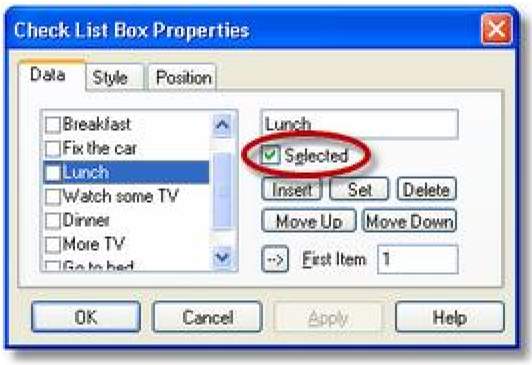
☐ Horizontal Scroll Bar
☑ Vertical Scroll Bar
☐ Wrap words
☐ Disabled
☑ Flat Style

Horizontal Offset 0
Horizontal Margin 2
Item Height 16
Horizontal Alignment Left
Vertical Alignment Centre

On the "Data" tab, delete the default list items and insert new items such as those given below. Type each one into the edit box then use the Alt+I shortcut to insert it into the list and keep the focus in the edit box ready to type the next item.

Walk the dog
Breakfast
Fix the car
Lunch
Watch some TV
Dinner
More TV
Go to bed

When you get to the "Lunch" item, check the "Selected" option. Any number of items may have the selected property but for the purposes of this tutorial, just leave this one selected:



☐ Breakfast
☐ Fix the car
☐ Lunch
☐ Watch some TV
☐ Dinner
☐ More TV
☐ Go to bed

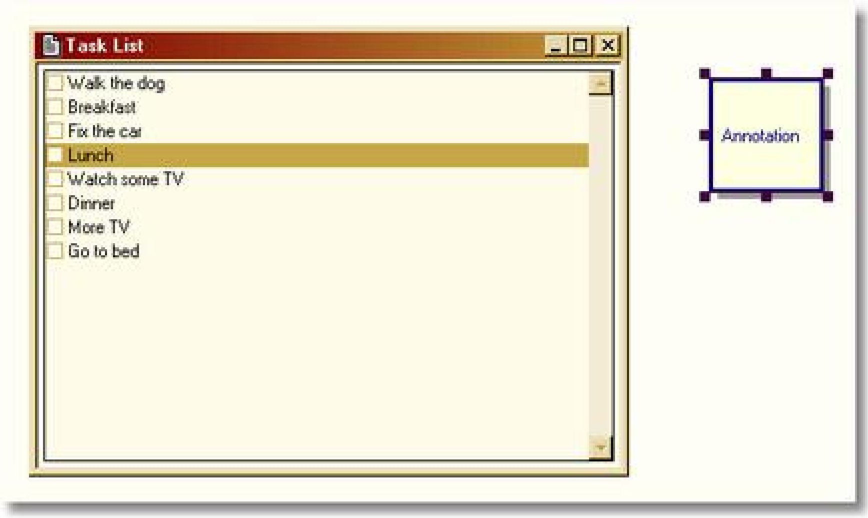
Lunch
☑ Selected
First Item 1

Hit "OK" to accept the changes. Note that although most changes appear on the design document as you make them or when you hit "Apply", the "Cancel" button can always be used to restore the element to the state it was in before the properties dialog was opened. Alternatively, you can accept the changes and use the Undo feature to restore the state (Ctrl+Z or "Edit | Undo" menu command).

6. The basic concept design for the Task List document window is complete but the real thing should probably have some more information in it so lets annotate the design to open up a discussion of requirements or just as a reminder.

   Annotations elements exist on another layer that floats above the design elements. They otherwise behave in the same way and may be selected along with design elements.

   Switch to the "Annotations" panel and add a "Boxed Annotation" to the right of the document window:



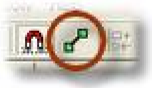
□ Walk the dog
□ Breakfast
□ Fix the car
□ Lunch
□ Watch some TV
□ Dinner
□ More TV
□ Go to bed

Edit the annotation properties and set the Title to "More Info" and enter something like the following for the Annotation text: "We probably want more info in columns with headers for sorting...".

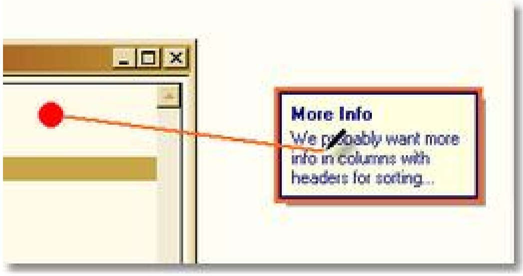
The box will need to be resized so that all of the text fits neatly.

7. You could leave it at that, or place the annotation box on top of the list, but to further the example, lets add a marker to connect the annotation box to.

   Place a "Large Dot Target" ● on the list. Now click on the "Make Connection" command on the toolbar to initiate connection mode:



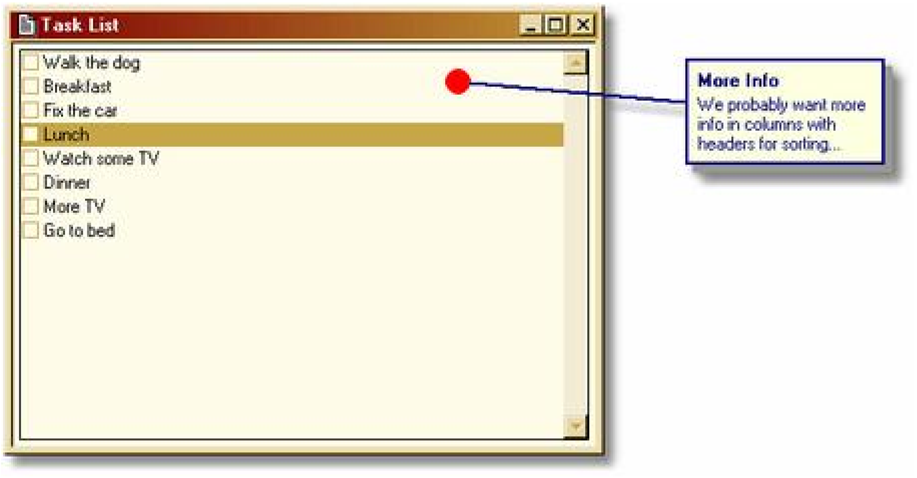
   The cursor will change to a pencil shape and an orange box will appear around anything that can be connected. Click on the dot target to start the connection then click on the annotation box to complete it:



More Info

We probably want more info in columns with headers for sorting...

8. The Document Window design is now complete so save your work.

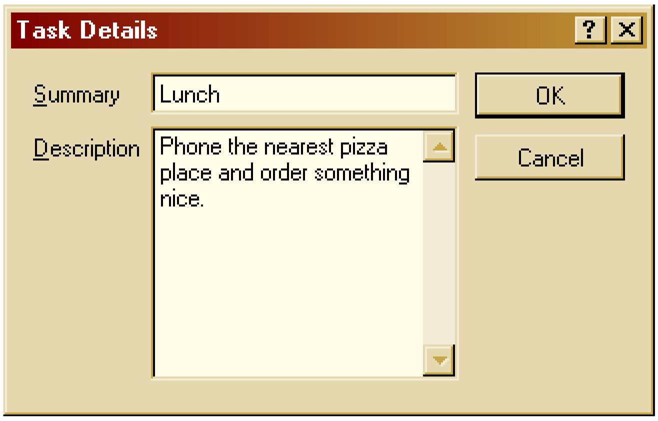


□ Walk the dog
□ Breakfast
□ Fix the car
□ Lunch
□ Watch some TV
□ Dinner
□ More TV
□ Go to bed

More Info
We probably want more info in columns with headers for sorting...

Step 5: Creating the Edit Task dialog

Tutorial Project File: EditTaskDialog.gui



Summary: Lunch

Description: Phone the nearest pizza place and order something nice.

The Edit Task dialog is to appear when the user wants to edit an existing task. It is identical to the New Task dialog except for the addition of some sample task details so we shall duplicate it rather than create it again from scratch.

We could just use the one dialog but it is useful when demonstrating conceptual designs to have a number of instances to illustrate different scenarios. The act of creating variations might also bring into question whether the dialogs should actually be the same or whether they need to be different.

1. Start this step by copying the existing New Task dialog design. Open the "NewTaskDialog.gui" file you created earlier by double-clicking it in the project tree.

   Use the "Edit | Select All" menu command to select all of the elements. Alternatively, use the Ctrl+A shortcut but you may need to click on the editor window first to give it the keyboard focus.

   Now use the "Edit | Copy" command (Ctrl+C) to copy all of the selected elements to the clipboard.

2. With the "Project" panel still open, create a new design document and call it "EditTaskDialog".

3. Now use the "Edit | Paste" command (Ctrl+V) to paste in all of the previously copied elements. Move them into the centre of the screen. Alternatively, use the "Paste" command from the right-click context menu to paste them at the mouse cursor location.

4. In keeping with the item we selected in the Task List document list in Step 4: Creating a Document Window, change the Summary edit box text to "Lunch" and enter something appropriate in the Description field such as: "Phone the nearest pizza place and order something nice."

5. That's this step complete so save your work.

The method above showed how to copy all (or part) of one design and place it in another. In fact, there is a much simpler way to achieve the result we need in this case without using Copy and Paste. The technique below will be used in Part 2 of this tutorial.

1. Open the "NewTaskDialog.gui" file by double-clicking it in the project tree (as above).

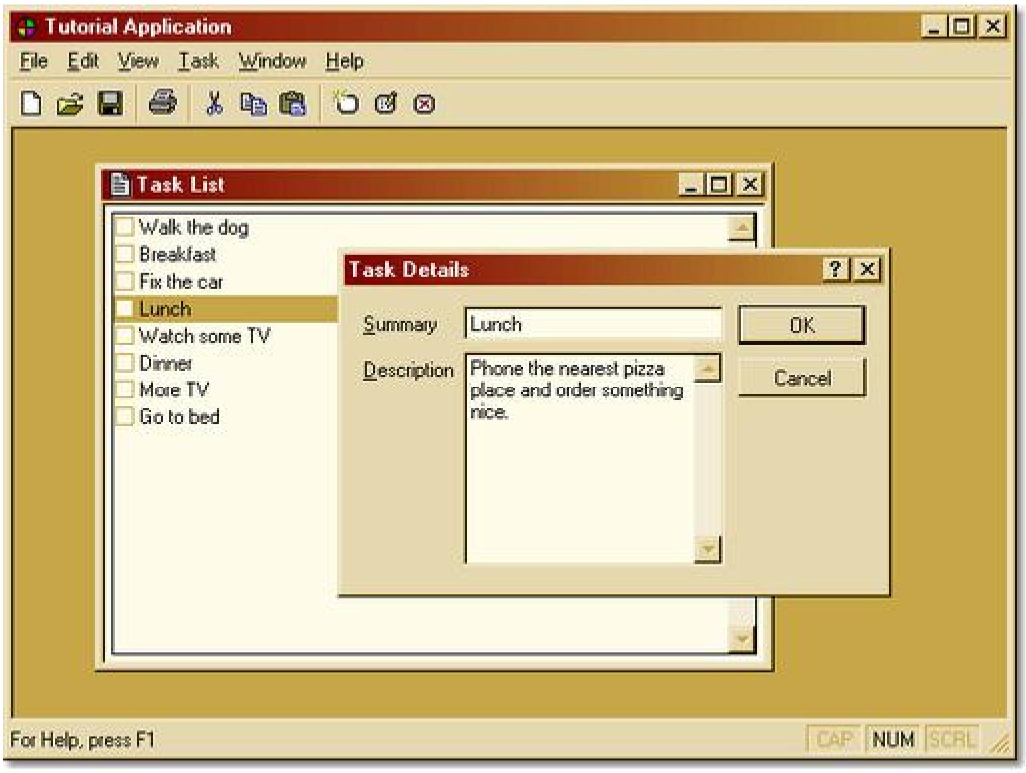
2. Use the "File | Save As..." menu command to save the design as a new file called "EditTaskDialog.gui". It will automatically appear and be selected in the Project file tree.

3. Now make the Summary and Description box changes as in 4 above.

4. Save your work.

Step 6: Putting together a screenshot

Tutorial Project File: Screenshot.gui



□ Walk the dog
□ Breakfast
□ Fix the car
□ Lunch
□ Watch some TV
□ Dinner
□ More TV
□ Go to bed

Summary: Lunch

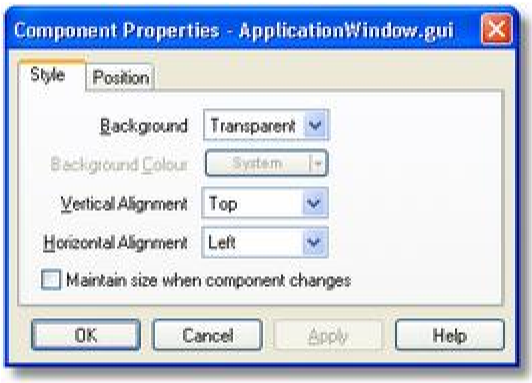
Description: Phone the nearest pizza place and order something nice.

To illustrate how the Task List Manager application might look during use, we can construct a screenshot out of the component parts.

1. Open the "Project" panel and create a new design document called "Screenshot".

2. Drag and drop the "ApplicationWindow.gui" design file from the project tree onto the design. Note that it can only be selected in its entirety; the individual elements that form the design are locked. Also note that the boxes that appear around the flat toolbar buttons in the original design to make them visible are hidden.

   If you open its property dialog you will see that it is a "Component" element with very few properties (you will not need to change any of these):



Background: Transparent

Background Colour: System

Vertical Alignment: Top

Horizontal Alignment: Left

□ Maintain size when component changes

   In fact, component elements only reference the associated design file rather than include a copy. That means that changes to the file will appear in the referencing design when it is re-opened or refreshed (F5 key).

3. In exactly the same way, add the "TaskListDocument.gui" and "EditTaskDialog.gui" design files as components to build up the screenshot (see the picture at the start of this step).

   Note that the lines of the Design Grid used on the Edit Task dialog are hidden when the design is used as a component. So too are the annotations on the Document Window.

   You may also have noticed that one consequence of using components rather than copies of the design elements is that the edge snapping is restricted to the outer edge of the component. So, for example, the inside, sunken frame of the Application Window could not be snapped to.

4. The screenshot is complete so save your work.

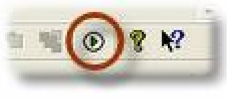
Step 7: Previewing the result

So far, all of your work has taken place in Design Mode using the design mode colour scheme set in the Preferences. Sometimes it is useful to get a quick preview of how a design would look with a different colour scheme (or your current Windows colour scheme) without any Annotations, Design Grid lines or visibility enhancements such as those on flat style Toolbar Buttons.

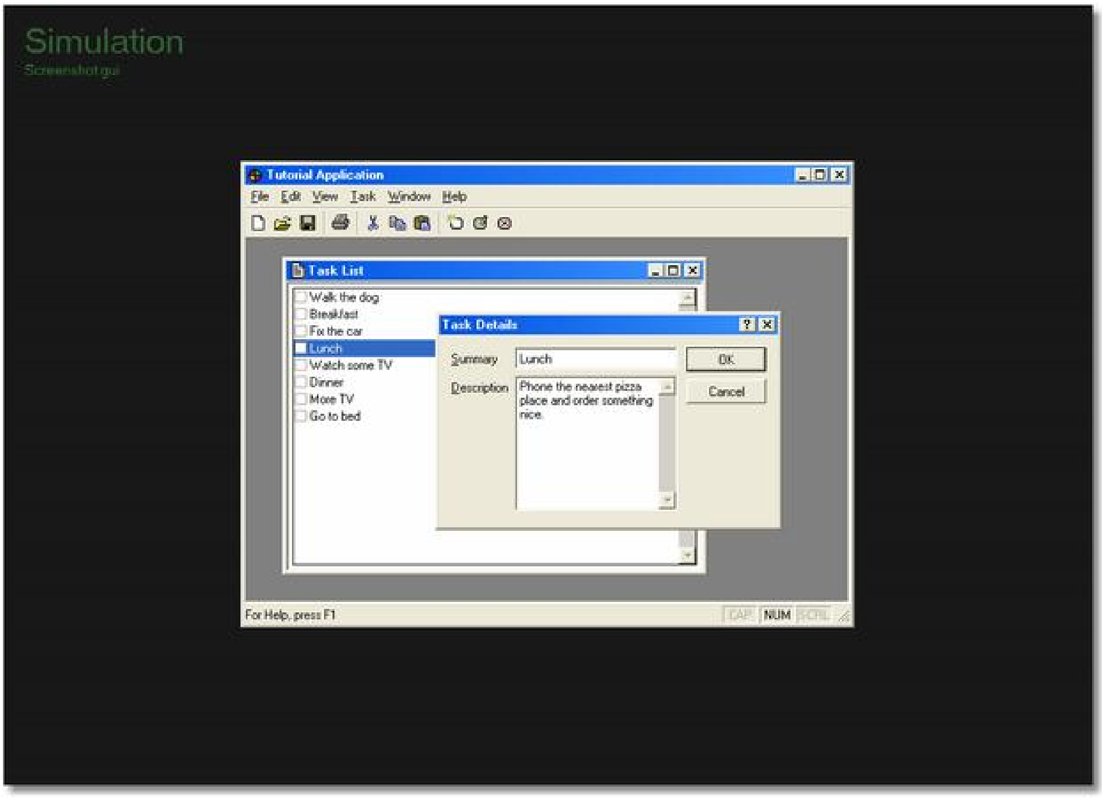
This is done using the Simulator which presents the design in a full screen window so that you are not distracted by other application windows. You will learn more about the Simulator later in this tutorial in Part 2: Bringing the design to life.

1. Open the "Screenshot.gui" file you were working on in Step 6: Putting together a screenshot (this will of course work with any design file though).

2. Use the "Simulator | Run Simulator" menu command (F9 shortcut key) or use the toolbar button:



The result should be something like that shown below. To exit simulation mode and return to the editor, you must press the **Escape** key.



- ☐ Walk the dog
- ☐ Breakfast
- ☐ Fix the car
- ☐ Lunch
- ☐ Watch some TV
- ☐ Dinner
- ☐ More TV
- ☐ Go to bed

Summary: Lunch

Description: Phone the nearest pizza place and order something nice.

For Help, press F1

Part 1 Summary

In Part 1 of this tutorial you have learnt how to:

- create a project (Step 1)
- create a design document (Step 2)
- place elements onto a design (Step 2)
- use automatic edge snapping (Step 2)
- use a design grid to create consistent spacing (Step 2)
- align and resize elements (Step 2)
- change element properties (Step 2)
- create application menu bars, toolbars and status bars (Step 3)
- place icons onto a design (Step 3)
- add design annotations (Step 4)
- duplicate designs for modification (Step 5)
- compose screenshots from multiple design components (Step 6)
- preview a design using the simulator (Step 7)

Part 2: Bringing the design to life

Part 1 showed you how to create component GUI parts and use them to compose a static screenshot. Part 2 will now show you how to turn your components into interactive application prototypes.

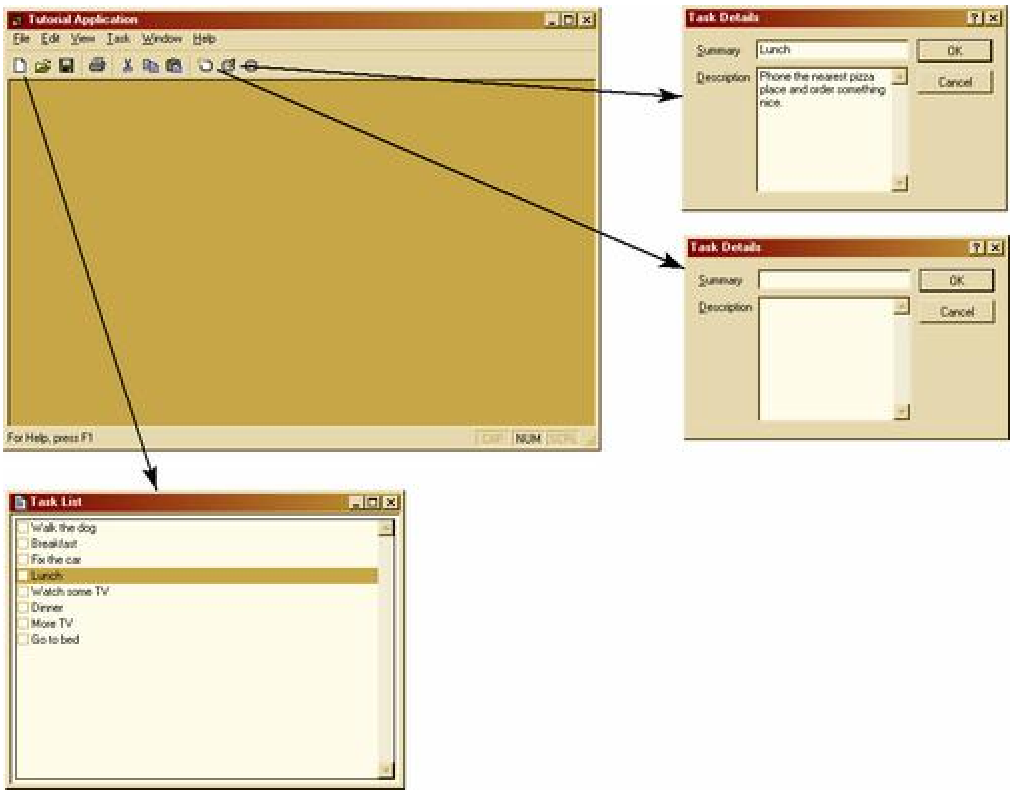
In the world of film, the term 'storyboard' is used to define a sequence of pictures that when put together convey a story line much the same way as a comic book. In software terms, a storyboard is a sequence of GUI screens that convey the flow of control as users take action. One key difference is that where film storyboards are strictly linear, software nearly always offers a choice of actions at each step leading to trees of workflows.

All of the components from Part 1 will be used here to build some storyboards. In addition, a new document window design will be created and a standard dialog component will be introduced.

Part 2 covers the essential steps required to create storyboards and interactive prototypes.

Step 1: Creating a simple storyboard

Tutorial Project File: Storyboard1.gui



Summary: Lunch

Description: Phone the nearest pizza place and order something nice.

- □ Walk the dog
- □ Breakfast
- □ Fix the car
- □ Lunch
- □ Watch some TV
- □ Dinner
- □ More TV
- □ Go to bed

In Part 1, Step 6: Putting together a screenshot you saw how to pull in component design elements to create a screenshot composition. We will use a similar technique here to put together a simple storyboard that shows program flow and how the user of our Tutorial Application can navigate between windows. We can then run this in the Simulator to demonstrate our design and evaluate usability issues.

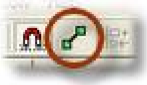
As a general tip while working on storyboards, you will probably want to get familiar with the commands and mouse actions for zooming in and out and panning around a design document (see the pop-up and reference help for details). Once you start laying out components and connecting them together, it will become impossible to see everything on screen at the same time at 100% zoom.

1. Open the "Project" panel and create a new design document called "Storyboard1".

2. Drag and drop the "ApplicationWindow.gui" design file from the project tree onto the design.

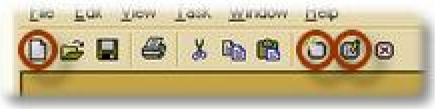
3. In exactly the same way, add the "TaskListDocument.gui", "EditTaskDialog.gui" and "NewTaskDialog.gui" design files and place them as shown in the picture at the start of this step. In contrast to creating a screenshot, for storyboards the window elements must be separated rather than placed on top of each other.

4. Now we simply use the "Make Connection" tool to create navigation paths in just the same way that the annotation elements were connected together in Part 1, Step 4:Creating a Document Window:

   Because we want to make more than one connection, this time hold the Shift key down when clicking on the toolbar button. Connection mode will then remain active after each connection has been completed, ready for the next one. Click the toolbar button again or press the Escape key to cancel connection mode when finished.

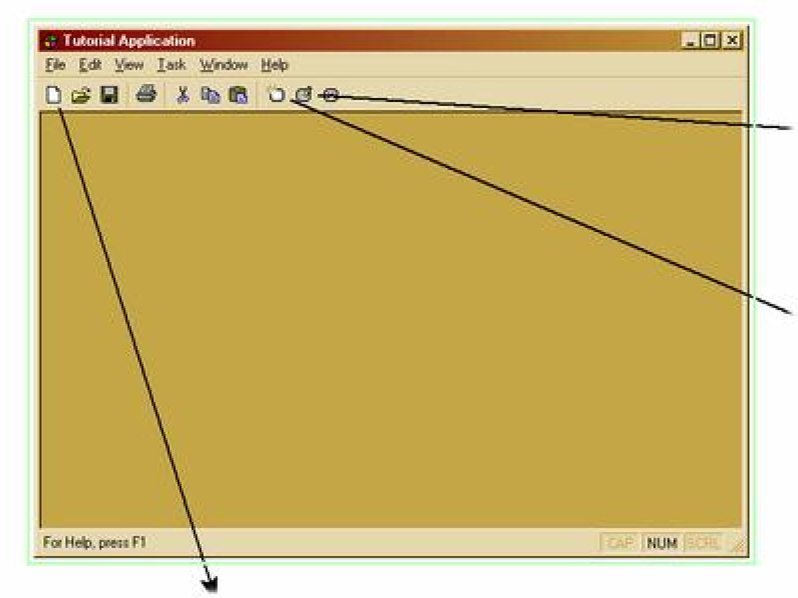
Connect the toolbar buttons shown below to the TaskListDocument, NewTaskDialog and EditTaskDialog components respectively.



Notice that although the ApplicationWindow is an atomic unit for selection and manipulation purposes, the sub-elements in the component (e.g. the buttons) are still accessible for making connections.

5. Save your work.

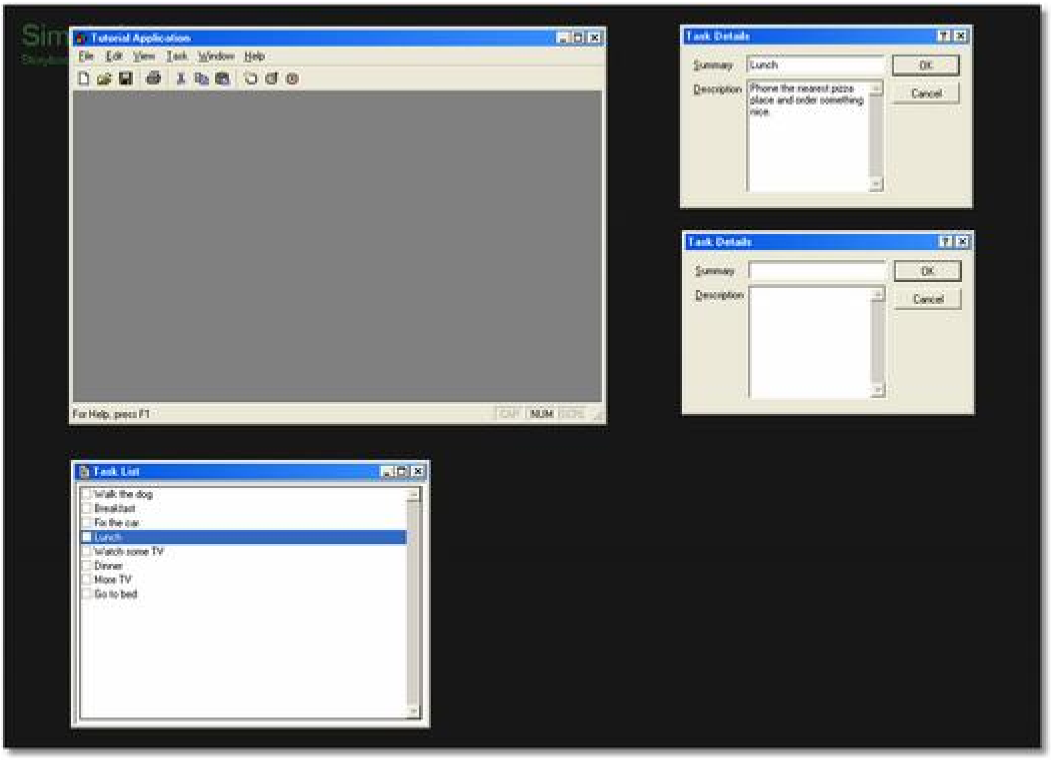
6. If you are using version 2.4 or above, you may notice a light green box around the application window. This means that it has been automatically selected as the starting point or 'representative element' for the design:



If you see the light green box then skip to item 9 below.

Otherwise, continue with the next item...

7. Now run the Simulator (F9 key) to see what happens:



Summary: Lunch

Description: Phone the nearest pizza place and order something nice.

Summary:

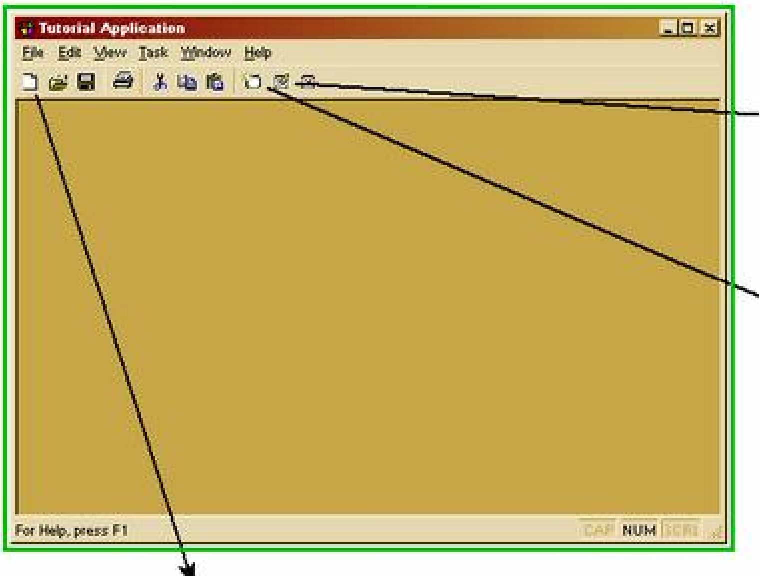
Description:

- □ Walk the dog
- □ Breakfast
- □ Fix the car
- □ Lunch
- □ Watch some TV
- □ Dinner
- □ More TV
- □ Go to bed

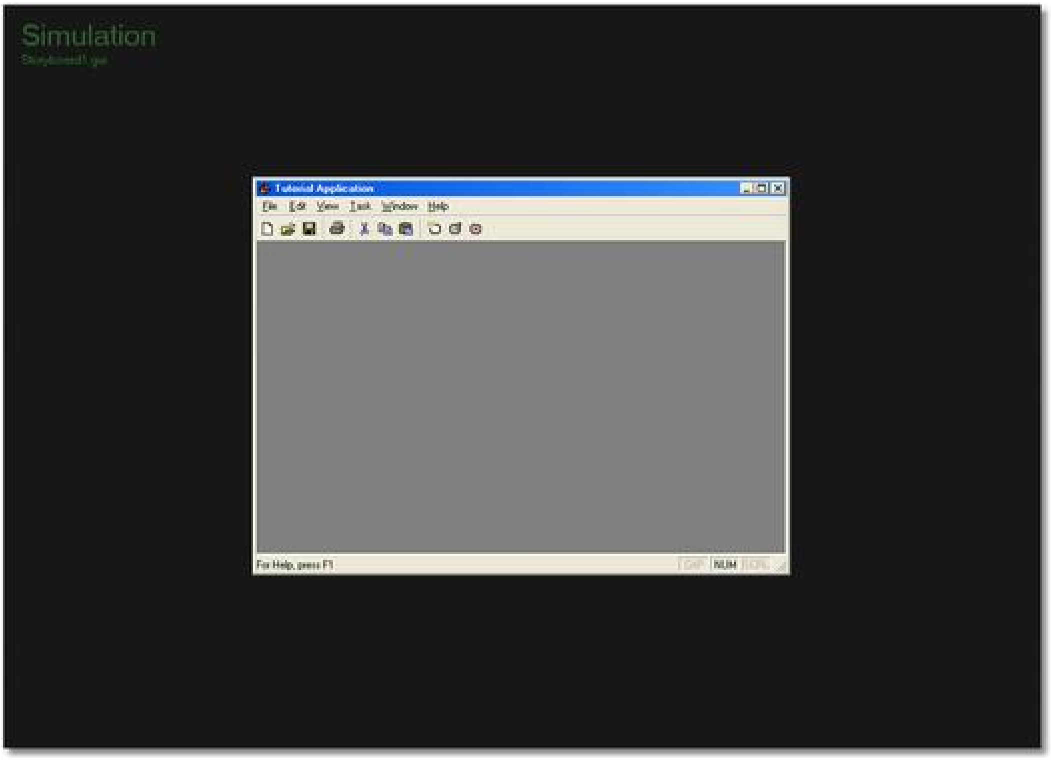
By default, the Simulator considers everything in the design to be part of one big screen. This gives you the flexibility to create any kind of screenshot or to just quickly throw some controls onto a design without worrying about the containing dialog and its size.

However, in this case we want to just see the application window, so...

8. Hit Escape to exit Simulation mode and select the ApplicationWindow component element. Now use the "Design | Select Representative Element" menu command (F10 shortcut key and also available from the right-click context menu). A green box will appear around the element to indicate that it has been chosen to represent the whole screen.



9. Now run the simulator again and this time all should be well:



As you hover the mouse over the buttons, the cursor changes to a hand when something is click-able (a bit like a hyperlink on a web page).

Click on the 'New Task' toolbar button to see the associated dialog appear. Just like a real dialog you can no longer access the windows below it. The toolbar buttons have become inactive. The dialog is a "modal" window. Within the Simulator, modal windows stack up on top of each other with only the top 'mode' active.

Each mode can be closed with the Escape key. When the last mode is closed the Simulator itself closes. If the modal stack is high and you want to exit the Simulator, you can either hit Escape repeatedly to close each mode or just hit Shift+Escape to get straight out.

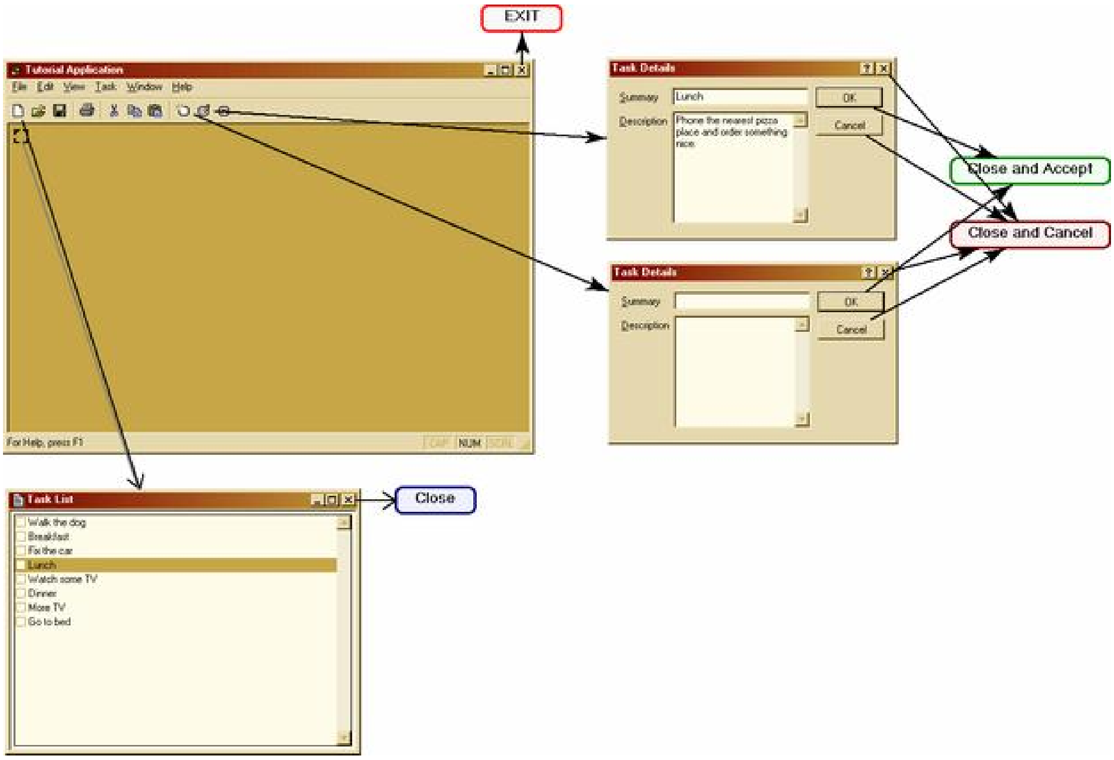
Hit the Escape key now to close the dialog then try again with the 'Edit Task' and 'New Document' toolbar buttons.

10. This step is now complete so save your work again.

In the next step you will see how to activate the "OK" and "Cancel" buttons, change the document window to appear as a non-modal or "modeless" window and control its positioning.

Step 2: Adding navigational features

Tutorial Project File: Storyboard2.gui



Summary: Lunch

Description: Phone the nearest pizza place and order something nice

Summary:

Description:

Walk the dog
Breakfast
Fix the car
Lunch
Watch some TV
Dinner
More TV
Go to bed

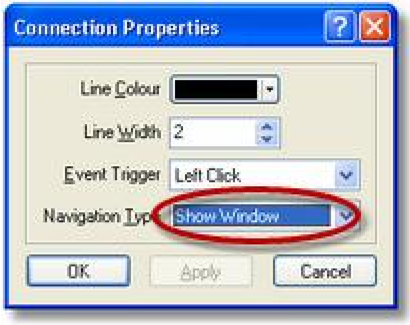
In this step we will start to improve the storyboard by changing the Task List document from a modal to a modeless window. We shall also activate the close buttons so that the dialogs and windows can be closed in a more natural way using the mouse rather than the Escape key.

Note that from version 2.0 onwards, "Close" boxes are not necessary in many situations because Frame Window "[X]" (close) boxes and certain Button elements automatically provide this behaviour without having to add the links. However, it is useful to know these techniques for situations where you *do* need to add closure controls manually.

1. Open the "Storyboard1" design document.

2. You would normally make gradual, incremental changes to a single storyboard to create an application prototype but for the purposes of this tutorial, we will separate each step into a separate file. Use the "File | Save As..." menu command to save the design as a new file called "Storyboard2.gui". It will automatically appear and be selected in the Project file tree.

3. The first change we want is to make the document window modeless. Double-click the connection line between the 'New Document' toolbar button and the Task List document window to open up its properties. Then change the "Navigation Type" from the default "Modal Popup" to "Show Window":



Line Colour

Line Width 2

Event Trigger Left Click

Navigation Typ Show Window

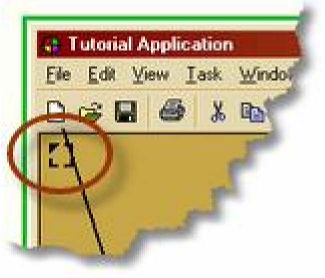
Hit "OK" to accept the change and you will notice that the arrow head has changed from a closed to an open shape.

4. Run the Simulator (F9) and check that after you press the 'New Document' toolbar button to open the Task List document window, you can now click on the 'New Task' or 'Edit Task' buttons to open one of the other dialogs. Clicking on the 'New Document' button again does nothing since the window is already open.

   Hit Escape to exit the Simulator.

5. By default, all new windows are opened in the centre of the screen. This isn't quite what we want for the document window so we need to add a positioning anchor.

   From the "Storyboard" panel on the Design Bar, drag and drop the small "Window Position Anchor" onto the application window where you want to place the document window:



   Now use the "Make Connection" command to link the anchor to the document window. Test it again in the Simulator.

6. While we're working on the document window, add a "Close" button from the "Storyboard" panel and connect to it from the 'X' close button on the title bar of the window as shown in the picture at the start of this step.

   Test this in the Simulator to see that the window closes and that you can re-open it with the 'New Document' toolbar button again.

   This step is unnecessary in version 2.0 onwards.

7. For the dialog "OK" and "Cancel" buttons, add a "Close and Accept" and "Close and Cancel" element from the "Storyboard" panel and connect to them from the buttons as shown in the picture at the start of this step. You can use as many of these elements as you wish but since the dialogs are so close together they can share them. Also connect up the 'X' close buttons on the title bars (should they cancel or accept?).

   If you test in the Simulator now you will find that the "OK" and "Cancel" buttons work as expected. In practice, there is no real difference between "Close and Accept" and "Close and Cancel". They both close the mode at the top of the stack and all windows associated with it (in this case just the dialog windows but if they were used instead of the "Close" element for the document window, the application window would close with it since they are at the same 'modal level'). By using different elements, the purpose of the buttons is made clear in the design. Also, future versions of GUI Design Studio may make a distinction so by using them in your designs now they will be ready for any new functionality.

   This step is unnecessary in version 2.0 onwards although it can make design intentions clearer.

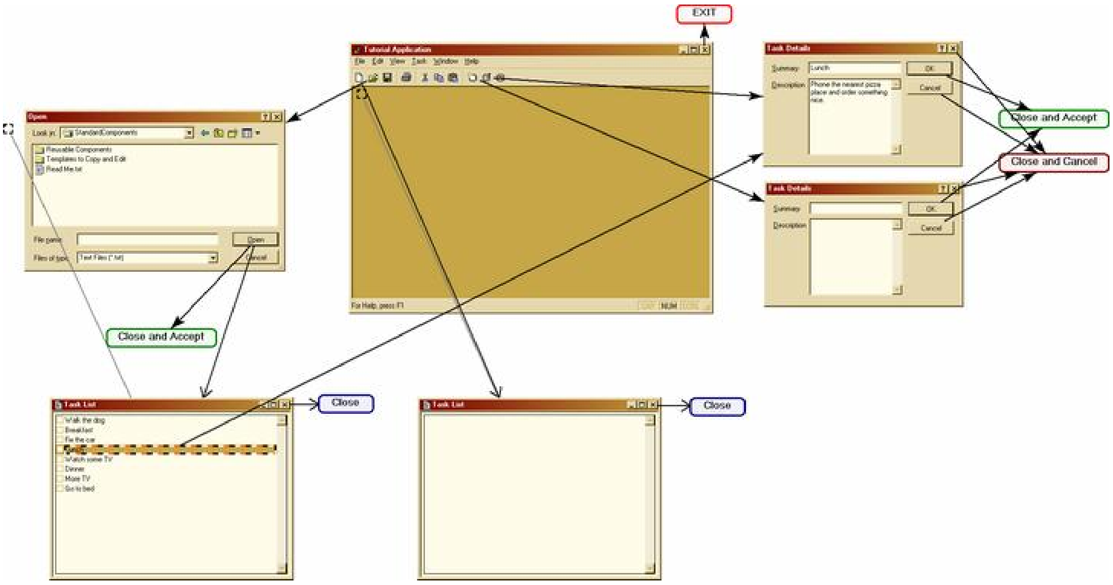
8. Finally add an "Exit" element and connect to it from the 'X' close button on the title bar of the application window as shown in the picture at the start of this step. Whenever you activate an "Exit" box, it closes the Simulator.

9. Test that everything works as you would expect in the Simulator and then save your work.

To understand a bit more about the differences between "Exit" and the various "Close" boxes, you might like to experiment by changing what the various close buttons connect to. For example, try connecting the application window's title bar 'X' button to a "Close" box instead of "Exit" and see what happens when you press it with and without the document window open.

Step 3: Adding a standard File Open dialog

Tutorial Project Files: EmptyDocument.gui
Storyboard3.gui



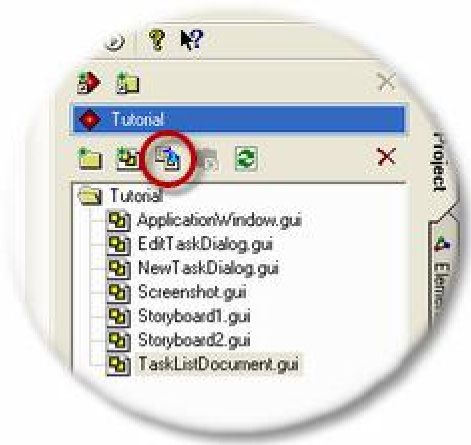
In this step we will build on the functionality of the previous step and introduce the use of standard library components. You will also see how navigational elements such as buttons can be made to perform multiple actions. We will refine the application prototype by trying to illustrate the difference between creating a new, empty document and opening an existing one.

If you are unsure of any of the steps, refer to the picture above for guidance.

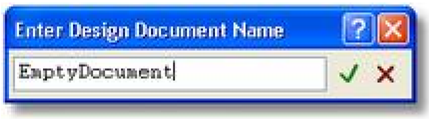
1. The first thing we need to do is to create a new "Empty Document" design file. We will use the existing "Task List Document" design as a base.

   Step 2: Adding navigational features showed one way of duplicating a design (open up the "TaskListDocument.gui" file then use the "File | Save As..." menu command to re-save it as "EmptyDocument.gui") but we will introduce a faster technique here.

   Simply select the "TaskListDocument.gui" file in the project tree then use the "Project | Duplicate Design..." menu option or associated Project panel command button:



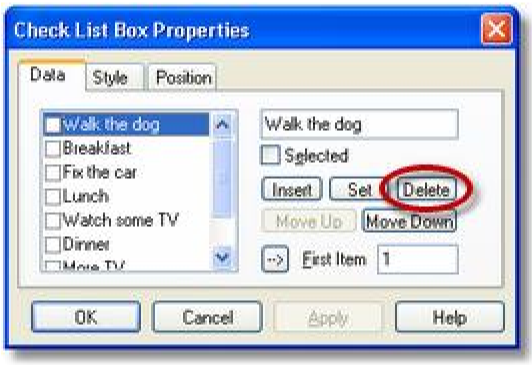
Now enter "EmptyDocument" as the name of the new document and it will be created and opened ready for editing:



EmptyDocument

2. Optionally, select the two annotation elements and delete them using the "Edit | Delete" menu command or the Delete key. Note that connections between elements are automatically deleted whenever either of the two elements they connect are deleted.

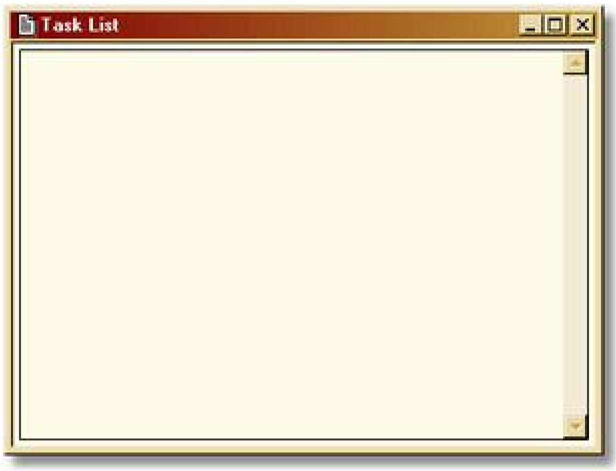
3. Double-click the task list box to open up its properties editor and "Delete" all of the list entries (you will need to select one first to enable the "Delete" command button):



□ Walk the dog
□ Breakfast
□ Fix the car
□ Lunch
□ Watch some TV
□ Dinner
□ More TV

Walk the dog
□ Selected
First Item 1

4. Hit "OK" to accept the changes and then **save the design**.



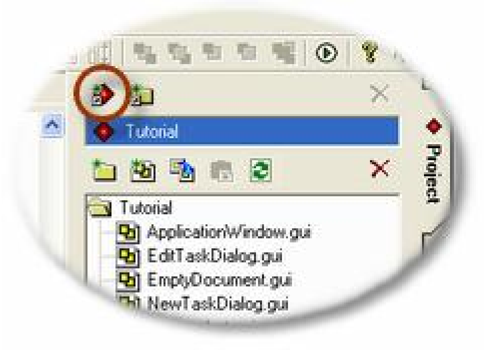
5. Now it's time to return to the storyboard. As before, for the purposes of this tutorial, we will save the results of each step to a separate design file rather than evolve a single storyboard file. So either use the Duplicate Design method or open the "Storyboard2" design document and use the "File | Save As..." menu command to create a new design file called "Storyboard3.gui".

6. Select the Task List document window and its associated Close button and move them to the left to make room for the new empty document window. Now drag and drop the "EmptyDocument.gui" design file you just created from the Project panel in its place.

7. We want to re-assign the functionality of the New Document command to open the empty document window. Select and delete the connection lines to the first document window from the command button and the positioning anchor. Now create new connections to the empty document window.

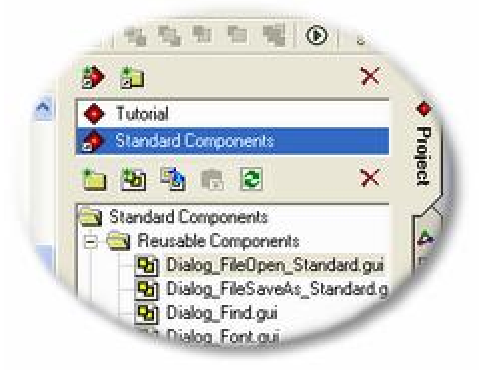
8. Add another "Close" box from the Storyboard panel and connect that to the empty document window's 'X' close button on its title bar. Use the Simulator (F9 key) to test the new behaviour.

9. The last component we need to add to the storyboard is a File Open dialog. This is a common requirement so a suitable example has been provided in the "Standard Components" library project. To access this we need to create a project link to it. On the Project panel, press the "Add Project Link" command button:



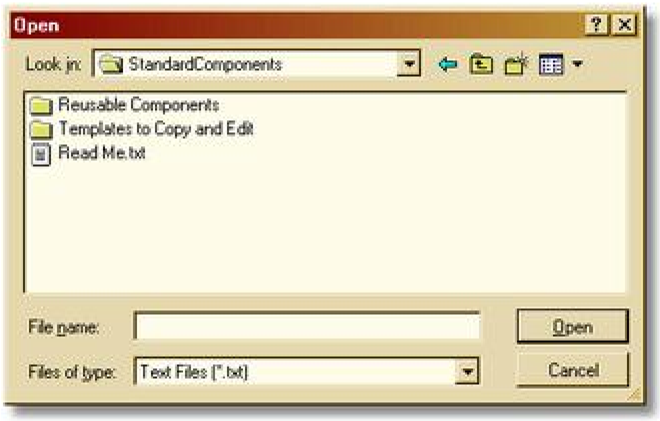
Browse for the "<GUI Design Studio install folder>\Samples\Standard Components\Standard Components.gdp" project file. This will be added to the Project/Folder list.

10. Locate the File Open dialog component file ("Dialog_FileOpen_Standard.gui"):



The standard File Open dialog contains example folder names and other details. These are unlikely to match your application requirements exactly. You can create your own dialog using the standard one as a template (just open it then save to a new file) or place other elements over the top to hide the parts you want to change.

For many purposes you may just want to use the standard dialog as-is simply to indicate that a File Open dialog will appear. That is what we shall do here so drag and drop the file as a component and place it to the right of the main application window.


Open
Look in: StandardComponents
Reusable Components
Templates to Copy and Edit
Read Me.txt
File name:
Files of type: Text Files (*.txt)
Open
Cancel

11. Open up the File Open file ("Dialog_FileOpen_Standard.gui") to examine the design by double-clicking it in the Project browser. You will see that it has its own set of navigations to close the dialog and open up a 'view options' drop-down menu, but when you dragged it onto your design as a component all you got was the dialog itself.

    This is a powerful feature of GUI Design Studio. It allows you to create and storyboard sub-parts of a project then reuse the whole thing as a component in another design where just the representative element is displayed.

12. We now need to hook up the File Open dialog. Start by adding a default Modal Popup connection from the Open button on the toolbar to the dialog.

    Test this with the Simulator (F9 key) and note that you can open the dialog and close it again with the "Open", "Cancel" and "X" close buttons. The View Options drop-down in the dialog should also appear when the associated button is clicked.

13. What we really want the "Open" button to do is close the dialog and then open up our Task List document window. Just as with everything else, you can access the buttons within the dialog component to create new navigations. Whenever you do this, the new navigation completely overrides any underlying navigation in the component design itself.

    When multiple navigation actions occur from a single button (or other source), they are executed in the order in which they were created. This is very important to understand, particularly in this case. If we show the document window before closing the dialog then the window will be at the same modal level as the dialog and will therefore close immediately with the dialog.

    First add a "Close and Accept" box from the Storyboard panel to the design and add a link to it from the "Open" button.

    Now add a further navigation link from the "Open" button to the Task List document window and change the Navigation Type to "Show Window".

    Test it all out in the Simulator (F9 key) to make sure it all behaves as expected. Note that the two document windows are independent of each other and may be individually opened and closed.

14. At this stage, the Task List document window will once again be appearing at its default location in the centre of the screen so we need to add another positioning anchor.

    [The important thing to note here though is that anchors position items relative to the calling window since there is no other suitable point of reference. This works out well in every case except the one we have here where we are showing a window on a different modal level after closing a dialog.

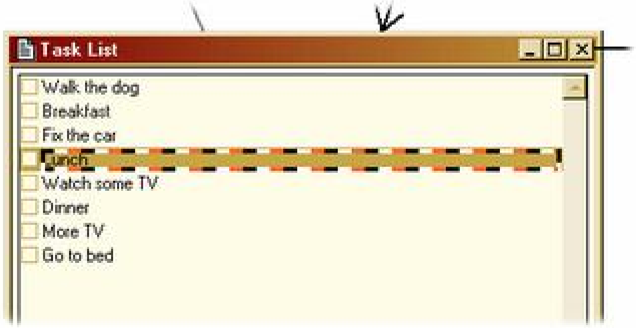
    If you place the anchor in the obvious place on the application window then the document window will appear way off to the right, possibly even off screen. Instead, the anchor needs to be placed relative to the File Open dialog. A good place is just to the left as shown in the picture at the start of this step. Unfortunately, you will need to experiment a bit to find the exact position that has the desired effect.]

ADDENDUM: As of version 1.0.0.34, the restriction set out in brackets above no longer applies. The tutorial project has not yet been updated and the existing offset anchor placement still works as before but you can now place anchors exactly where you would expect to. In this case, the anchor can be placed on the application window.

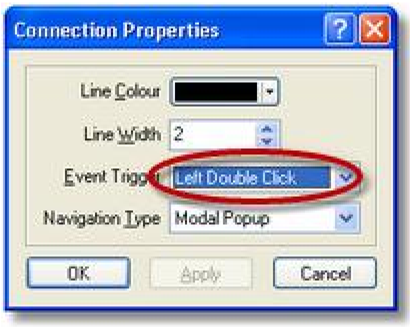
15. Finally, to wrap up this step, we'll look at overlay elements and see how they can be used to create navigation actions from parts of elements that normally wouldn't source them such as bitmapped screenshot images.

    In this case, we will set up a link so that when the user double-clicks the selected (highlighted) item in the Task List document window, it will open up the Task Details editor dialog as an alternative to hitting the 'Edit Task' toolbar button.

    On the Storyboard panel, locate the striped box "Navigation Overlay" element. Drag this onto the design and position it over the highlighted task as shown below:



    Now add a connection from the overlay box to the filled in Edit Task Details dialog already on the design. Double-click the connection line to open its property editor and change the "Event Trigger" option from the default "Left Click" to "Left Double Click":

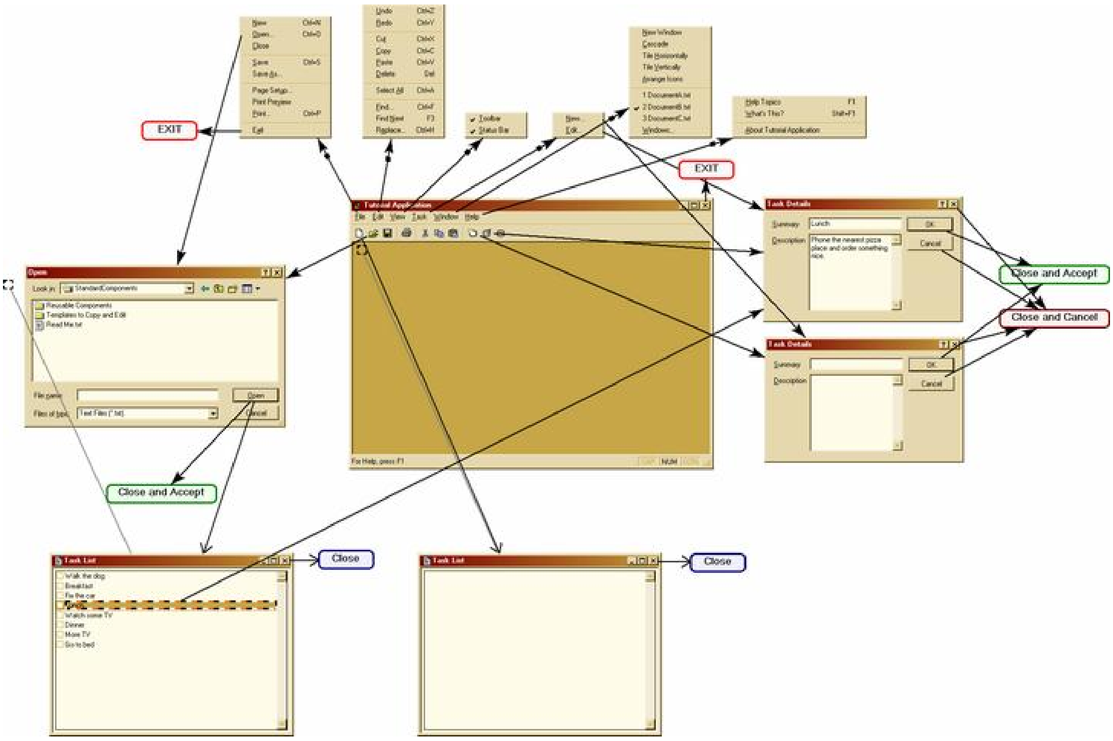


Line Colour
Line Width 2
Event Trigger Left Double Click
Navigation Type Modal Popup

16. As always, test that everything works as you would expect in the Simulator and then save your work.

Step 4: Adding a menu system

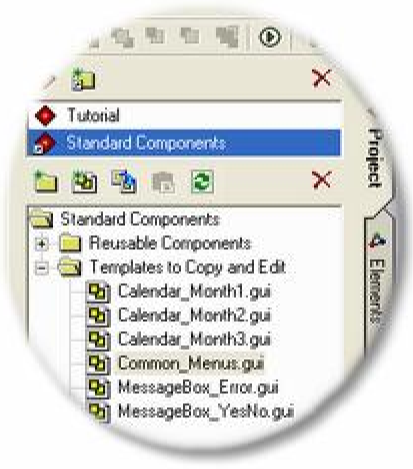
Tutorial Project File: Storyboard4.gui



In this final step we will complete the design by building a simple menu system.

1. Use your preferred method ('Duplicate Design' or 'Open and Save As') to create a new "Storyboard4.gui" design document as an exact copy of "Storyboard3".

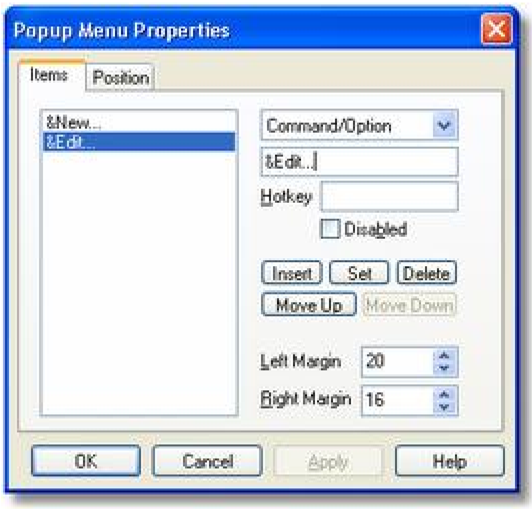
2. Each of the popup menus could be created from scratch but to save time we shall use some standard menu elements as a starting point. Open the Common Menus component file ("Common_Menus.gui") from the "Standard Components" library project:



3. Use Ctrl + Left Click to select the File, Edit, View, Window and Help menu elements then press Ctrl + C to copy them to the clipboard.

4. Switch back to the "Storyboard4" design document and use Ctrl + V to paste them in. Now arrange them above the Application Window in the design as shown in the picture at the start of this step. If you want to, you can use the align tools to make them look neat.

5. We still need a custom "Task" menu so add a new "Popup Menu" element from the "Toolbars and Menus" category on the Elements panel. Edit its properties to provide "New..." and "Edit..." entries:



&New...
&Edit...

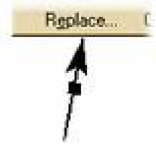
Command/Option
&Edit...
Hotkey
□ Disabled

Left Margin 20
Right Margin 16

If you want to, you can also change the "About Application Name" item in the Help menu to "About Tutorial Application".

6. Connect each of the items on the Application Window's menu bar to its respective menu. Since there are a few to do, remember to use the Shift key when clicking the "Make Connection" toolbar button to make the command sticky. Click the button again or use Escape to cancel the mode when finished.

   Because the targets are popup menus, you will notice that the Navigation Type on each of the connections is automatically set to "Modal Choice Popup" as indicated by the square box on the arrow line:



   This is a special type which causes the target window (in this case a menu) to close if the user clicks outside of the window. It also allows cascading menus to close properly.

7. Add an "Exit" box from the Storyboard panel and connect to this from the Exit command on the File menu.

8. Connect the "Open..." command on the File menu to the File Open dialog.

9. Connect the "New..." and "Edit..." commands on the Task menu to the associated Task Details dialogs.

10. Test everything out in the Simulator (F9 key) and save your work.

Part 2 Summary

In Part 2 of this tutorial you have learnt how to:

- create a storyboard (Step 1)
- select a representative element for a design document (Step 1)
- make connections between appropriate elements (Step 1)
- navigate between windows in a design and in the Simulator (Step 1)
- make a window appear modal or non-modal (Step 2)
- position windows using anchors (Step 2)
- create navigation actions that close windows (Step 2)
- create navigation actions that close the Simulator (Step 2)
- add links to library projects (Step 3)
- access components and elements in library projects (Step 3)
- override navigation actions in components (Step 3)
- use overlays to add navigation to elements that wouldn't normally navigate (Step 3)
- change the event that triggers navigation (Step 3)
- add a menu system (Step 4)

Congratulations on reaching the end of this tutorial!

Index