Basic Dialog with Tree and List

The *Basic Dialog with Tree and List* pattern creates a Dialog Window with a series of controls including a Tree control and List control and three buttons that can be used to visualize the user interface of a Windows application, to generate the code to an application's Resources file for compilation or to simulate its states with a behavioral diagram using JavaScript.

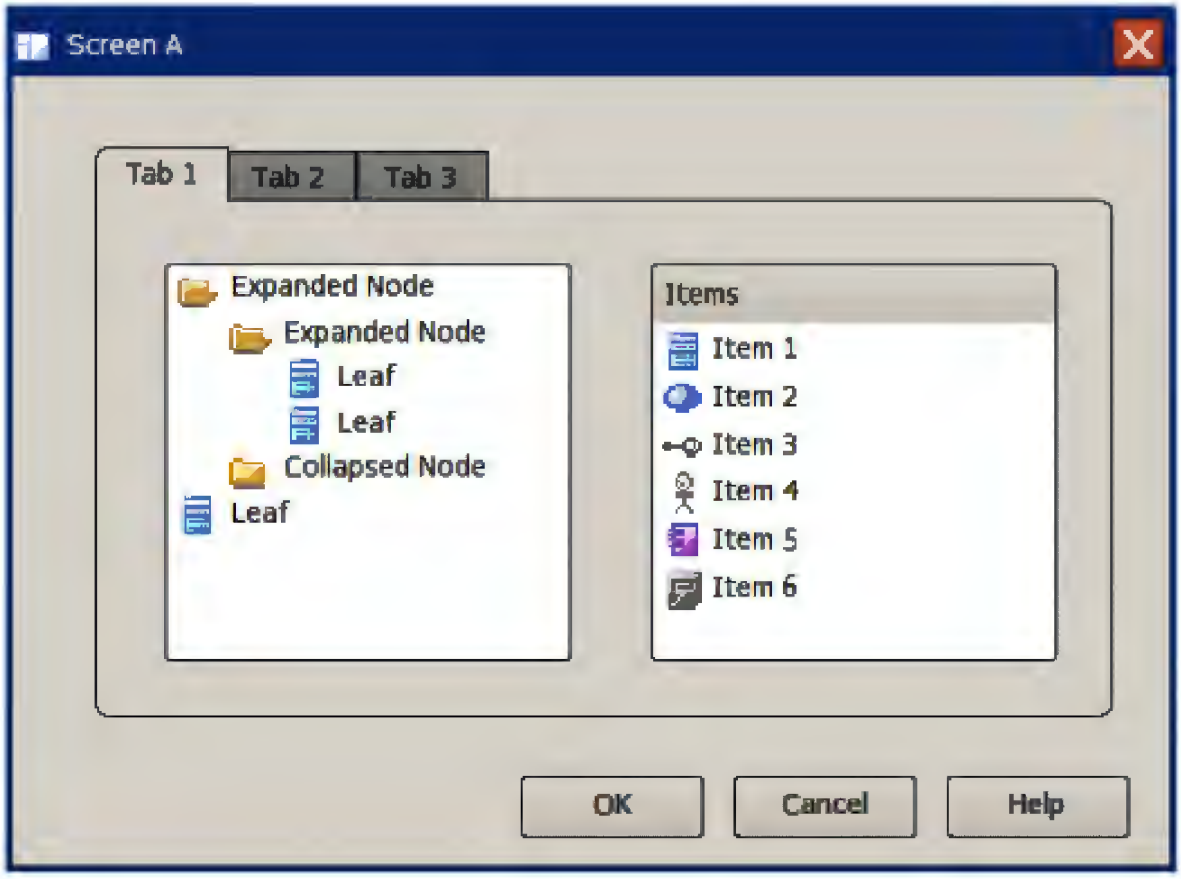

Figure 1. Shows a Window 32 Bit User Interface diagram with a simple Dialog Window with a Tree View, List View and three buttons: OK, Cancel and Help.

Discussion

The purpose of the pattern is to allow technical Ux Designers, software engineers and other stakeholders to create and view Windows 32Bit dialog boxes that can be used to visualize a user interface and as inputs into the implementation of an application. The dialog can been designed with a Tree Control that would be typically be used as a navigation for a container node (such as a directory) that when selected would display its contents (such as files) in the List control. The buttons would be used to submit the Dialog, Cancel the Changes or Open help. The controls can all have their Win32 properties set directly in the application.

The dialog is useful in the following situations:

- Visualize a dialog as part of a User Interface for a Win32 application.
- Get sign off and feedback from users to ensure the design meets their requirements and expectations.
- Use in conjunctions with behavioral models such as State machines to which can be used to trigger changes in the dialog using JavaScript commands.

The following is a list of some things you may want to do when working with this pattern.

- Change the name of the Package and the diagram to suit the initiative.
- Change the name of the Dialog and the Controls to suit the initiative.
- Set the properties of the dialog and the each control using the Win32 section of the Properties window.
- Create additional controls, lay them out and set their properties.

The following is a list of some of the next steps available when applying the pattern.

Create a behavioral model and use JavaScript to create Simulations of the dialog.

Export the Dialog to a Win32 Project Resource file (*.rc) or synchronize it with an existing file.

Reference

The following help topics will assist you learn about how to work with this pattern.

Design a New Dialog

Win32 UI Technology

Export Dialog to RC File

Import All Dialogs from RC File

The following are some of the tools that will be helpful when working with this pattern.

The following are some of the tools that will be helpful when working with this pattern.

Source Code Editor

The Source Code Editor is a fully featured programming source code editor. It has a structure tree for easy navigation of attributes, properties and methods. Line numbers can be displayed and syntax highlight options can be configured. Many of the features that software engineers are familiar with in their favorite IDE, such as Intelli-sense and code completion are included in the editor. Viewing the source code juxtaposed with the Models from which it is generated brings a great clarity to the design effort and its implementation. For more details see the Editing Source Code help topic.

Visual Execution Analyzer

The Visual Execution Analyzer is made up of an advanced and powerful suite of tools that allow you to build, debug, record, profile, simulate and otherwise construct and verify your software development while keeping the code tightly integrated with your model. Enterprise Architect has rich support for a wide range of popular compilers and platforms, in particular the Java, .Net and Microsoft Windows C++ environments. Software development becomes a highly streamlined visual experience, quite unlike working in traditional environments. For more details see the Visual Execution Analyzer help topic.

Model Transformation

The Model Transformation facility allows a modeler to transform a Conceptual data model to a Logical data model and in turn a Logical Data Model to a Physical Data Model. The transformations are driven by user-defined or built-in templates. This facility

will save time and effort and reduce the possibility of errors. For more details see the Model Transformation help topic.

Hand Drawn and Whiteboard Diagrams

The Hand Drawn and Whiteboard Mode are display options available for any diagram that changes a system-drawn diagram to appear as though it was drawn by hand and, optionally, hand drawn on a whiteboard. It is a powerful device to engage an audience by presenting the diagram in a rough and more immediate style giving the impression that it is just a sketch that can be changed. For more details see the Hand Drawn and Whiteboard Mode help topic.

Alternate and Images for Diagram Elements

Most standard elements allow an alternate image to be defined for an element that will be used in place of the graphical notation for the element either on a selected diagram or as a default on all diagrams. For more details see the Using the Image Manager help topic.

Pan and Zoom

The Pan and Zoom facility is one of the tools that can be used to navigate around a large diagram. Often the resolution of a diagram must be reduced to ensure it is wholly visible but by using the Pan and Zoom window you can leave the diagram at a readable resolution and pan around to areas of interest zooming in when necessary. For more details see the Pan and Zoom help topic.

Document Generator

The Document Generator is a powerful facility in Enterprise Architect that allows a Database Engineer or other stakeholder to create high quality corporate or technical documentation directly from the model, suitable for internal or external audiences. For more details see the Documentation help topic or the more general topic on Model Publishing.