User Manual for wxBuilder 2.0

Julian Smart
Decision Support Group
Artificial Intelligence Applications Institute
80 South Bridge
University of Edinburgh
EH1 1HN

March 1995

Contents

1. Introduction

wxBuilder is a tool for interactively building the GUI (Graphical User Interface) component of a wxWindows application. The user chooses and places frames, dialog boxes, subwindows and panel items, and wxBuilder generates appropriate C++ code for compiling with the wxWindows library on Windows or X. Optionally, the wxWindows resource file format (suffix .WXR) may be generated to help separate out GUI specification from the application code.

The user may then populate this skeleton program with further code to complete the application, or use the results as a design and prototyping aid. wxBuilder could also find use as a learning aid for a beginning wxWindows user, since there is a clear correspondence between GUI and C++ code.

wxBuilder may also be used as a migration tool from straight Windows to wxWindows, by importing existing Windows resource (.RC) files (Windows version only).

wxBuilder is not unique in the kind of facilities it provides, but it is the only known freely available tool to generate multi-platform code, and is the only one to support wxWindows.

1.1. Status of wxBuilder

wxBuilder now compiles under Windows and Motif. An XView release is available but there are problems related to multiple modal dialogs that will be fixed shortly.

Under Motif, there is a problem with the 'simulated' windows not always being drawn correctly; this is a tricky Motif-related problem. The current work-around is to move the simulated frame or dialog box, and the display will be refreshed correctly.

Various facilities are planned for a later release, including:

- one source file per window option
- MDI support
- command line parser generation
- help file generation
- wxCLIPS generation
- extend repertoire of GDI objects, such pens, brushes and colours.
- Support for wxForm

1.2. Change log

Version 2.0 November 1995

- Improved code for resource generation (was very buggy)
- WinHelp manual now has Win95 .cnt contents file

Version 1.9 March 1995

- Abandoned the simulation window and now use the features added in 1.61 (c) to directly manipulate panel and dialog items.

- The tree (for one frame or dialog only) is displayed in the main window.

Version 1.8 January 1995

- Major alterations to generate .WXR resource files (optionally).
- All elements that use bitmaps and icons now have a bitmap properties editor, which centralizes the bitmap code and allows specification of desired bitmap format on each platform.
- Added support for bitmap message.
- Fixed some bugs e.g. crashes when a toolbar is present.
- On loading a project, the main window is displayed immediately.

Version 1.7 November 1994

- Added support for wxGauge and wxGroupBox.
- Improved sizing code.
- Fixed bug in panel window style (was always wxBORDER).

Version 1.6 October 1994

- Added support for wxSlider.
- Changed 'int' window styles to 'long'.
- Added support for bitmap buttons.

Version 1.5 October 1994

- Cured bug in conversion of Windows menu resource to wxWindows menu (menu items lost their parents).
- First release of Motif version, with more stable Motif wxWindows implementation.
- Added ability to load bitmaps into canvases, for more detailed 'simulation' of interfaces.
- Added provision for extra (user-defined) library and include paths.
- Improved UNIX and DOS makefile generation.
- Solved several bugs (e.g. right-click editing of a window caused a crash).

Version 1.3 June 1994

- File history now has names added to front of list.
- Added Cancel button to on-exit dialog box.
- Fixed delete bug wxRadioBox/wxListBox/wxChoice properties dialogs.
- Fixed bug which disabled loading on the first attempt.
- Fixed code generation bugs for X/Windows icons.
- Fixed OnSize code generation bug (statements in wrong order).
- Fixed UNIX makefile generation bugs.
- Windows version no longer uses CTL3D since it seems to crash wxWindows programs on some PCs. Now uses CTL3DV2 instead.

Version 1.2 May 1994

- Added file history.
- Some code generation bugs fixed.

2. Creating a project with wxBuilder

This chapter guides the user through the main processes involved in creating a new application.

2.1. Running wxBuilder

Run `wxbuild.exe` (Windows) or `wxbuild` (UNIX) with an optional project filename. Project files contain all information relevant to an application except bitmap files and icons; their extension is `.WXP` (for wxWindows Project).

Below is a screen shot of the main wxBuilder window, with a project already loaded. A menu bar is supplemented by a tool bar to accelerate certain commands. The *object editor* canvas displays a tree representing the hierarchy of windows for the current frame or dialog box.

Below the object editor canvas is a listbox containing a list of all top-level windows (all the frames and dialog boxes). Clicking on a window in this listbox creates the window and updates the object editor.

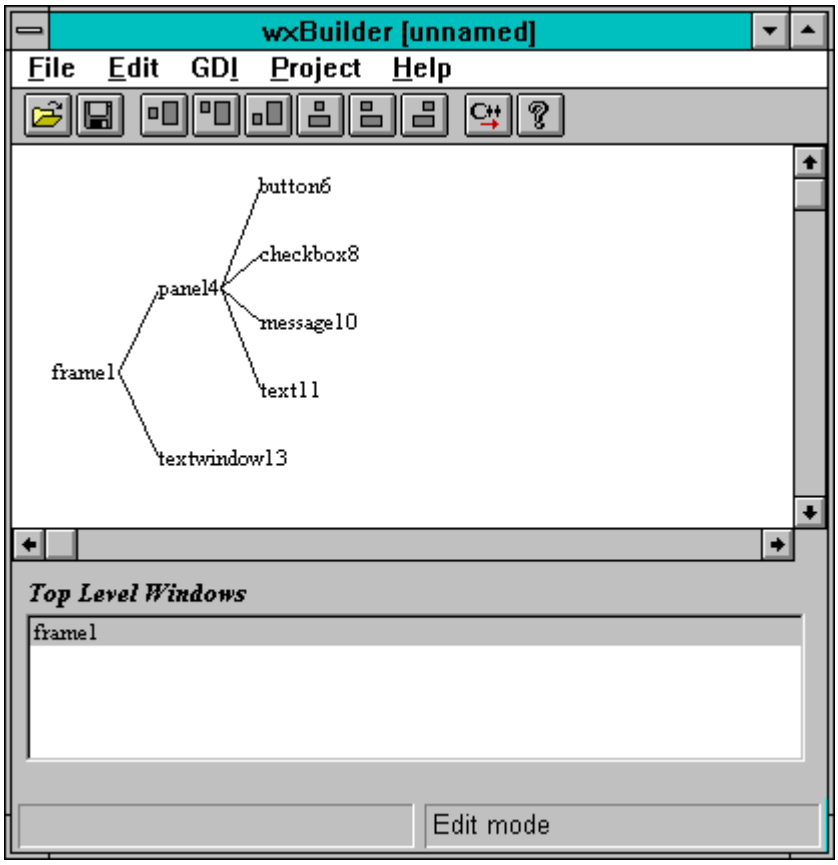

Figure 1: Main wxBuilder window

The status line is divided into two fields: the left field displays help messages, and the right field shows the current 'mode', either edit or test mode. In test mode, certain events from project windows (such as pressing buttons or selecting menu items) will do the user-defined action, whereas in edit mode, the same events will pop up a dialog to allow editing of that action.

An *object palette* is displayed in a sepearate frame. This allows selection of a window object (such as a canvas or button) -- the cursor in the object editor window changes to a cross, and left-clicking on a panel or dialog box (in edit mode) creates an item of that kind. Items other than panel items must be created from the Edit menu at present, since only panels and dialog boxes are sensitive to left clicks. This will probably change.

Figure 2: Object palette

2.2. Setting up for a new project

It's wise to set things up properly before starting to create windows. The **Project** menu has two items you should look at: **Edit project settings** and **Edit global settings**.

Project settings are saved with the project file and are changed on a per-project basis. These include project name (used as a base for generating files), and project directory name.

Global settings are variables that are likely to change on a per-user basis, and are stored in WIN.INI (under Windows) or .Xdefaults (under X). Under X, the .Xdefaults file should be edited by hand, since the X versions of wxBuilder do not currently save global settings. See *Resources* (page 19).

In addition, set the name of the application class (to be used as a derivative of wxApp) and a brief description of the project, using the **Edit application properties** item on the **File** menu.

2.3. Creating a main window

Every wxWindows must have a frame which is the main window (in C++, returned from the **wxApp::OnInit** function).

The first frame you create is taken to be this main window. Choose **Edit: New frame** or click on the frame tool on the palette, and then on the object editor.

You can size and position this frame using the representation on the object editor (try to resist the temptation to edit the real windows directly!). See *Using the object editor* (page 6) for further details.

3. Procedures

3.1. Changing project and global settings

3.1.1. Project settings

Project settings are saved with the project file and are changed on a per-project basis.

Setting	Description
Root name	This will be used for all files generated by wxBuilder for this project. Extensions such as .RC, .CC and .DEF will be appended to form filenames.
C++ extension	Allows you to choose what C++ extension will be used (default .CC).
Directory under Windows	The directory into which the generated files will be placed under Windows.
Directory under UNIX	The directory into which the generated files will be placed under UNIX.
Generate makefiles	If on, will generate makefiles.
Generate RC file	If on, will generate Windows resource file.
Generate DEF file	If on, will generate Windows module definition file.
Generate wxWindows resources	If on, will generate .WXR resource files (different from Windows resource files), replacing the programmatic creation of menu bars, dialog boxes and panels where possible.

3.1.2. Global settings

Global settings are variables that are likely to change on a per-user basis, and are stored in WIN.INI (under Windows) or .Xdefaults (under X). Under X, the .Xdefaults file should be edited by hand, since the X versions of wxBuilder do not currently save global settings.

These are the global settings.

Setting	Description
Windows compiler	Set to the desired Windows compiler. At present, the only recognised compilers are Microsoft's C++ version 7, Visual C++ and the NT compiler. For other compilers, please write the makefile by hand or use a Microsoft compatibility mode if available.
UNIX compiler	Type in the name of the UNIX compiler. The default is gcc-2.1.
X GUI target	UNIX makefiles are generated with three targets, allowing simultaneous compilation of XView , Sun Motif and HP Motif binaries. This target specifies which one should be used for auto-compiling *(not yet implemented).*
X includes	Switches specifying where to find X include directories.

X libs Switches specifying where to find X library directories.

Extra includes (UNIX) Allows specification of extra include paths for UNIX makefiles.

Extra libs (UNIX) Allows specification of extra library paths for UNIX makefiles.

Extra includes (Windows) Allows specification of extra include paths for Windows makefiles.

Extra libs (Windows) Allows specification of extra library paths for Windows makefiles.

UNIX wxWindows directory wxWindows home directory under UNIX.

Windows wxWindows directory wxWindows home directory under Windows.

See also *Resources* (page 19).

3.2. Using the object editor

The object editor is a hierarchical view of the windows in your frame or dialog box.

To edit a window, you may create a new frame or dialog box from the Edit menu, or click on a window name in the listbox on the main wxBuilder window

Panel items may be repositioned by dragging with the left mouse button, selected and deselected with left click, and resized by dragging on the handles visible when the object is selected. Frames may be sized interactively as normal, and dialog box size may be specified in the relevant properties dialog.

Each type of window has a property editor associated with it, allowing tailoring of specific window characteristics. To invoke a property editor, right-click on the panel item or object in the hierarchy. This pops up a menu for editing or deleting the object.

To align panel items, select two or more items, and use an alignment tool on the tool bar. The first item selected is taken as the reference for aligning the other items.

3.3. Generating C++

Click on the **C++** tool on the tool bar, or choose **Generate C++** from the **Project** menu. A report window will be displayed with error and warning messages, and if successful, source files will be written in the current project directory.

3.4. Managing GDI objects

wxBuilder allows the user to define a set of GDI (Graphics Device Interface) objects to be used in a program. Currently only fonts are supported, but in future this will be extended to colours, pens, brushes, bitmaps, icons and cursors.

To define a set of fonts for the project, invoke the Font Manager from the **GDI** menu.

Quit Quit from the Font Manager.

Help Invoke help for the Font Manager.

Add Add a font.

Delete Delete the selected font.

3.5. Creating a frame

Left-click on the frame icon on the object palette, and left-click again on the object editor canvas, or choose **New frame** from the **Edit** menu. The frame properties editor will be shown, with the following controls.

Name The name of this frame instance.

Description A textual description of the frame.

Class name The name of the C++ class that will be created for this frame. At present, each frame must have a unique class.

Icon name The name of the icon file to be used for this frame.

Title The default title for the frame.

No. status line fields The number of divisions for the status line. Zero indicates no status line.

Vertical subwindow tiling If checked on, the *subwindow tiling* (page 7) is vertical. Otherwise it is horizontal.

Thick frame If checked on, the frame has a thick border for resizing (Windows only).

Caption If checked on, the frame has a caption (Windows only).

System menu If checked on, the frame has a system menu (Windows only).

Minimize button If checked on, the frame has a minimize button (Windows only).

Maximize button If checked on, the frame has a maximize button (Windows only).

Shuffle subwindows If pressed, the subwindow positioning order is rearranged.

3.5.1. Subwindow tiling

The way in which subwindows are positioned and sized in wxWindows is very different from the method for panel items, dialog boxes and frames. This is because subwindows are often sized relative to their parent frame.

If there is only one subwindow, the default wxWindows method is used, which is simply to resize the subwindow to the frame client area when the frame is created or resized.

If there is more than one subwindow, wxBuilder uses an algorithm to layout the windows. It is hoped that this algorithm will cater for most subwindow layout needs. These are the options and constraints:

1. Zero or more subwindows may be children of a frame.

2. All subwindows must be tiled either vertically or horizontally. Mixture of the two tiling modes is not allowed (horizontal tool bars work independently and may be discounted from this constraint).

3. Each subwindow has a *resize strategy*, one of Fixed, Proportional and Grow.

4. The mix of resize strategies must be such as to allow wxBuilder to determine all subwindow sizes.

If the resize strategy is Fixed, the specified width (if in horizontal tiling mode) or height (if in vertical mode) is used. The other dimension is determined by the size of the frame (and possibly the presence of a tool bar).

If the resize strategy is Proportional, the free dimension of the subwindow is determined from that dimension of the frame, together with the percentage specified in the subwindow property editor.

If the resize strategy is Grow, the free dimension of the subwindow is determined by adjacent subwindows. In practice only one subwindow may use this strategy.

3.6. Creating a dialog box

Left-click on the dialog box icon on the object palette, and left-click again on the object editor canvas, or choose **New dialog** from the **Edit** menu. The dialog box properties editor will be shown, with the following controls.

Name The name of this dialog box instance.

Description A textual description of the dialog box.

Class name The name of the C++ class that will be created for this dialog box.

Title The default title for the dialog box.

Width The width of the dialog box.

Height The width of the dialog box.

Fit contents If checked on, the panel will fit itself to its contents.

Horizontal labels If checked on, the default labelling orientation is horizontal, otherwise it is vertical.

Label font The name of the font to be used for panel item labels. The font should have been created previously using the font manager, accessible from the GDI menu.

Button font The name of the font to be used for panel item contents.

3.7. Creating a menu bar

To create a menu bar for a frame displayed in the object editor, select the **Edit menu bar** option on the **Edit** menu.

The Menu Bar Editor will appear with a large listbox for menu and item names, fields for editing text, and buttons below it for creating and deleting items.

Menus List of menus and menu items.

Name Contains the name of a new or currently selected menu item.

Id Allows the user to specify the name of the menu item identifier, as used in a **wxMenu::Append** call. These names will be associated with integers automatically by wxBuilder. If the user leaves this field blank, wxBuilder will generate a suitable identifier.

Help string Allows the user to specify a help string which will appear on the status line when the mouse cursor is over that menu item (Motif and Windows only).

New item Creates an item at the same level as the currently selected menu item. So, if the currently selected item is a menu called *Edit*, pressing this button will create a new menu, not an item on the *Edit* menu. Note that you must fill in the fields *before* you press the button.

New child Creates a *child* of the currently selected menu item, i.e. it creates a new submenu. So to create the first item of the *Edit* menu, select the *Edit* item and press this button. Note that you must fill in the fields *before* you press the button.

New separator Creates a separator after the selected menu item.

Delete item Deletes the currently selected menu item.

Save item Saves the displayed values in the currently selected menu item.

3.7.1. Hotkeys

To specify a hotkey, insert an ampersand (&) before the letter which serves as the hotkey (Windows and Motif). For example, **E&xit**.

3.7.2. Deleting a menu bar

To remove the entire menu bar, delete the object in the usual way from the object editor (select the menu bar object and choose the **Delete object** option from the **Edit** menu).

Alternatively, delete all the menus from the menu bar editor.

3.8. Creating a tool bar

To create a tool bar for a frame displayed in the object editor, select the **Edit tool bar** option on the **Edit** menu.

The tool bar properties editor will appear with the following controls.

Name The name of this tool bar instance.

Description A textual description of the tool bar.

Class name The name of the C++ class that will be created for this tool barl.

Rows/cols The maximum number of columns for this toolbar (set to a high number since

only one row is allowed).

Tools A list of the current tools in the tool bar.

Id Contains the name of the identifier for a new or currently selected tool. wxBuilder allocates actual integer identifiers for these names automatically.

Help string Allows the user to specify a help string which will appear on the status line when the mouse cursor is over that menu item (Motif and Windows only).

Bitmap The name of the bitmap file for the tool. A full path or extension should not be given; the project path will be prepended, and a .BMP or .XBM extension added as appropriate.

Toggle If checked on, this tool will be a toggle tool.

Add Press to add a tool to the tool bar. Note that you must fill in the fields *before* you press the button.

Save Saves the displayed values in the currently selected toggle.

Delete Deletes the currently selected tool.

Demote Moves the selected tool to the end of the tool bar.

3.8.1. Deleting a tool bar

To remove the entire tool bar, delete the object in the usual way from the object editor (select the tool bar object and choose the **Delete object** option from the **Edit** menu).

3.9. Creating a panel subwindow

Left-click on the panel icon on the object palette, and left-click again on the object editor, inside a frame object. The panel properties editor will be shown, with the following controls.

Name The name of this panel instance.

Description A textual description of the panel.

Class name The name of the C++ class that will be created for this panel.

Border If checked on, the panel will have a border.

Resize strategy One of Fixed, Proportional and Grow. See *subwindow tiling* (page 7).

Width The width of the panel.

Height The width of the panel.

% frame If Resize strategy is Proportional, this value specifies the proportion of the frame for determining the subwindow size.

Fit contents If checked on, the panel will fit itself to its contents.

Horizontal labels If checked on, the default labelling orientation is horizontal, otherwise it is vertical.

Label font The name of the font to be used for panel item labels. The font should have been created previously using the font manager, accessible from the GDI menu.

Button font The name of the font to be used for panel item contents.

3.10. Creating a canvas subwindow

Left-click on the canvas icon on the object palette, and left-click again on the object editor, inside a frame object. The canvas properties editor will be shown, with the following controls.

Name The name of this canvas instance.

Description A textual description of the canvas.

Class name The name of the C++ class that will be created for this canvas.

Border If checked on, the canvas will have a border.

Resize strategy One of Fixed, Proportional and Grow. See *subwindow tiling* (page 7).

Width The width of the canvas.

Height The width of the canvas.

% frame If Resize strategy is Proportional, this value specifies the proportion of the frame for determining the subwindow size.

Retained canvas If checked on, the canvas will be retained under X.

Pixels/unit X The number of pixels per logical X unit. If zero, no horizontal scrollbar will be shown.

Pixels/unit Y The number of pixels per logical Y unit. If zero, no vertical scrollbar will be shown.

No units X The number of logical units in the X dimension.

No units Y The number of logical units in the Y dimension.

Units/page X The number of logical units per horizontal page.

Units/page Y The number of logical units per vertical page.

Use bitmap If on, will load a bitmap into the canvas.

Edit bitmap properties Allows the user to change the name of the bitmap, and the method of bitmap creation on each of the X and Windows platforms.

3.10.1. Displaying a bitmap

A Windows bitmap (or GIF under X) may be displayed in the canvas. For code generation

purposes, this facility should only be used if the wxImage library (X) or DIB library (Windows) is available at your site.

3.11. Creating a text subwindow

Left-click on the text subwindow icon on the object palette, and left-click again on the object editor, inside a frame object. The text subwindow properties editor will be shown, with the following controls.

Name The name of this text subwindow instance.

Description A textual description of the text subwindow.

Class name The name of the C++ class that will be created for this text subwindow.

Border If checked on, the text subwindow will have a border.

Resize strategy One of Fixed, Proportional and Grow. See *subwindow tiling* (page 7).

Width The width of the subwindow.

Height The width of the subwindow.

% frame If Resize strategy is Proportional, this value specifies the proportion of the frame for determining the subwindow size.

Text file The name of a text file to be loaded onto the canvas on creation.

3.12. Creating a button item

Left-click on the button icon on the object palette, and left-click again on the object editor inside a panel or dialog box object. The button properties editor will be shown, with the following controls.

Name The name of this button.

Description A textual description of the button.

Label Label for the button.

Auto size If checked on, the button will size itself according the the text label. If off, the size will be determined by the interactively sizing the button from the object editor.

Use bitmap If on, will use a bitmap instead of a string label.

Edit bitmap properties Allows the user to change the name of the bitmap, and the method of bitmap creation on each of the X and Windows platforms.

3.13. Creating a checkbox item

Left-click on the checkbox icon on the object palette, and left-click again on the object editor inside a panel or dialog box object. The checkbox properties editor will be shown, with the following controls.

Name The name of this checkbox.

Description A textual description of the checkbox.

Label Label for the checkbox.

Auto size If checked on, the checkbox will size itself according the the text label. If off, the size will be determined by the interactively sizing the checkbox from the object editor.

Label position Horizontal, vertical or default label placement. Under Windows the checkbox label is always horizontal.

3.14. Creating a choice item

Left-click on the choice item icon on the object palette, and left-click again on the object editor inside a panel or dialog box object. The choice item properties editor will be shown, with the following controls.

Name The name of this choice item.

Description A textual description of the choice item.

Label Label for the choice item.

Auto size If checked on, the choice item will size according to wxWindows defaults. If off, the size will be determined by interactively sizing the choice item from the object editor.

Label position Horizontal, vertical or default label placement.

Values A list of default string values.

Value Text field for entering a choice item value.

Add Add the text in the current field to the list of default values.

Delete Delete the currently selected item.

3.15. Creating a gauge item

Left-click on the gauge icon on the object palette, and left-click again on the object editor inside a panel or dialog box object. The gauge properties editor will be shown, with the following controls.

Name The name of this gauge.

Description A textual description of the gauge.

Label Label for the gauge.

Auto size If checked on, the gauge will size according to wxWindows defaults. If off, the size will be determined by interactively sizing the gauge from the object editor.

Label position Horizontal, vertical or default label placement.

Range Range of the gauge (default 100).

3.16. Creating a groupbox item

Left-click on the groupbox icon on the object palette, and left-click again on the object editor inside a panel or dialog box object. The groupbox properties editor will be shown, with the following controls.

Name The name of this groupbox.

Description A textual description of the groupbox.

Label Label for the groupbox.

Label position Horizontal, vertical or default label placement.

3.17. Creating a listbox item

Left-click on the listbox icon on the object palette, and left-click again on the object editor inside a panel or dialog box object. The listbox properties editor will be shown, with the following controls.

Name The name of this listbox.

Description A textual description of the listbox.

Label Label for the listbox.

Auto size If checked on, the listbox will size according to wxWindows defaults. If off, the size will be determined by interactively sizing the listbox from the object editor.

Label position Horizontal, vertical or default label placement.

Values A list of default string values.

Value Text field for entering a listbox value.

Add Add the text in the current field to the list of default values.

Delete Delete the currently selected item.

3.18. Creating a message item

Left-click on the message item icon on the object palette, and left-click again on the object editor inside a panel or dialog box object. The message item properties editor will be shown, with the following controls.

Name The name of this message item.

Description A textual description of the message item.

Label Label for the message item.

Use bitmap If on, will use a bitmap instead of a string message.

Edit bitmap properties Allows the user to change the name of the bitmap, and the method of bitmap creation on each of the X and Windows platforms.

3.19. Creating a radiobox item

Left-click on the radiobox icon on the object palette, and left-click again on the object editor inside a panel or dialog box object. The radiobox properties editor will be shown, with the following controls.

Name The name of this radiobox.

Description A textual description of the radiobox.

Label Label for the radiobox.

Auto size If checked on, the radiobox will size according to wxWindows defaults. If off, the size will be determined by interactively sizing the radiobox from the object editor.

Label position Horizontal, vertical or default label placement.

Rows/cols Number of rows or columns (depending on label orientation value).

Values A list of default string values.

Value Text field for entering a radiobox value.

Add Add the text in the current field to the list of default values.

Delete Delete the currently selected item.

3.20. Creating a slider item

Left-click on the slider icon on the object palette, and left-click again on the object editor inside a panel or dialog box object. The slider properties editor will be shown, with the following controls.

Name The name of this slider.

Description A textual description of the slider.

Label Label for the slider.

Auto size If checked on, the slider will size according to wxWindows defaults. If off, the size will be determined by interactively sizing the slider from the object editor.

Min Value Minimum integer slider value

Max Value Maximum integer slider value

The current position of the slider will be used as the default value.

3.21. Creating a text item

Left-click on the text item icon on the object palette, and left-click again on the object editor inside a panel or dialog box object. The text item properties editor will be shown, with the following controls.

Name The name of this text item.

Description A textual description of the text item.

Label Label for the text item.

Auto size If checked on, the text item will size according to wxWindows defaults. If off, the size will be determined by the interactively sizing the text item from the object editor.

Label position Horizontal, vertical or default label placement.

Value Text field for the default text value.

3.22. Creating a multi-line text item

Left-click on the multitext item icon on the object palette, and left-click again on the object editor inside a panel or dialog box object. The multitext item properties editor will be shown, with the following controls.

Name The name of this multitext item.

Description A textual description of the multitext item.

Label Label for the multitext item.

Auto size If checked on, the multitext item will size according to wxWindows defaults. If off, the size will be determined by the interactively sizing the multitext item from the object editor.

Label position Horizontal, vertical or default label placement.

Value Text field for the default text value.

3.23. Editing a window object's properties

An object's properties may be edited by one of three methods.

1. Select the object with left click on the object editor, and choose **Delete object** from the **Edit** menu.

2. Right-click on the object in the object editor.

3. Left-click on the object name in the object hierarchy window.

3.24. Deleting a window object

An object may be deleted by selecting it with left click on the object editor, and choosing **Delete object** from the **Edit** menu.

3.25. Associating actions with events

In order to give the interface some dynamics, a small repertoire of actions is available for association with some events. For example, a menu command called **Open report window** might be associated with with the **Open window** action, to open another frame in the interface.

To create an action, execute the event you wish to associate with an action. These events are currently:

- Menu commands

- Button depression
- Tool depression

You will be presented with a list of possible actions. Once you have chosen an action, the appropriate action editor is invoked.

The next time you invoke the event in Edit mode, the action editor will be presented. You may delete the action using the **Delete** button. In Test mode, the event will cause the appropriate action to be executed.

Currently, only one action per event is allowed. A later release of wxBuilder may allow multiple events, so (for example) an event could cause a window to be opened *and* a file to be loaded.

Note that some actions are mandatory for a working C++ program to be generated. For example, a modal dialog box with no action dismissing it will cause the window to be uncloseable, which might require the user to quit the windowing system to recover.

3.26. Converting Windows resource files

Menu bars and dialog boxes can be converted from Windows resource files, using Petr Smilauer's resource parser (Windows version of wxBuilder only). Menu bars convert well, but due to the differences between wxWindows panel items and Windows controls, dialog boxes convert less well.

Use the **Load Windows resources** option from the **Edit** menu. This opens a dialog box from which .RC files may be loaded, or added to the existing database of resources. Once the resource file has been loaded, quit from this dialog and use the **Convert menu bar** or **Convert dialog** option from the **Edit** menu.

4. Code generation details

Some or all of the following files may be generated:

- a header file (.H)
- a main source file (.CC)
- UNIX and DOS makefiles
- a module definition file (.DEF)
- a Windows resource file (.RC)

For every frame, dialog box and subwindow, a class is generated, to allow customization for each window instance except for panel items.

Frames, dialog boxes and subwindows are generated with data members for their contained windows. Pointers to all frame objects are held in the wxApp-derived class.

The description property is used to document class declarations and definitions.

If the user has specified actions for menu commands, buttons or tool bar tools, the appropriate wxFrame::OnMenuCommand members functions, wxToolBar::OnLeftClick member functions, or button callbacks are generated with working code, otherwise empty stubs (or case statements) are generated for the user to fill in.

Platform-specific resource loading is dealt with using conditional compilation statements.

5. Resources

The following is a list of the WIN.INI or .Xdefaults resources associated with wxBuilder. Under Windows, the resources follow the `[wxBuilder]` section in WIN.INI. Under X, each resource name should be prefixed by `wxBuilder.` in the .Xdefaults file.

Resource	Description
autoCompile	1 or 0.
autoRun	1 or 0 (not implemented).
compilerDOS	The name of the compiler in the DOS/Windows environment.
compilerUNIX	The name of the compiler in the UNIX environment.
xIncludes	List of X include directories.
xLib	List of X include directories.
extraIncludesX	List of extra include directories (under UNIX).
extraLibsX	List of extra library directories (under UNIX).
extraIncludesMSW	List of extra include directories (under Windows).
extraLibsMSW	List of extra library directories (under Windows).
wxDirUNIX	wxWindows directory under UNIX.
wxDirDOS	wxWindows directory under DOS.
mainX	Main wxBuilder window X coordinate.
mainY	Main wxBuilder window Y coordinate.
mainWidth	Main wxBuilder window width.
mainHeight	Main wxBuilder window height.
reportX	Report window X coordinate.
reportY	Report window Y coordinate.
reportWidth	Report window width.
reportHeight	Report window height.
treeX	Tree window X coordinate.
treeY	Tree window Y coordinate.
treeWidth	Tree window width.
treeHeight	Tree window height.

paletteX	Palette X coordinate.
paletteY	Palette Y coordinate.

Index