



ЭНЕРГЕТИКА
МИКРОЭЛЕКТРОНИКА
АВТОМАТИКА

РАСПРЕДЕЛЕННАЯ СИСТЕМА ДИСПЕТЧЕРСКОГО УПРАВЛЕНИЯ

Графический редактор интерфейсов
LabWindows/CVI

Руководство пользователя

РАСПРЕДЕЛЕННАЯ
СИСТЕМА
ДИСПЕТЧЕРСКОГО
УПРАВЛЕНИЯ



Содержание

Информация о данном руководстве	4
Организация справочного руководства	4
Условные обозначения, использованные в настоящем руководстве	5
Связь с заказчиком	5
Глава 1. Концепции пользовательского интерфейса	6
Введение в Графический пользовательский интерфейс	6
События пользовательского интерфейса	7
Управление пользовательским интерфейсом с помощью обратных вызовов или GetUserEvent	7
Связь с помощью исходного кода	8
Управляющие режимы для генерации событий Commit (исполнения)	8
Использование CodeBuilder (Построителя кода) для создания исходного кода Вашего GUI	9
Работа Графического пользовательского интерфейса	10
Использование панелей	10
Использование линеек меню	11
Использование элементов управления	12
Типы данных элементов управления	12
Цифровые элементы управления	13
Строчные элементы управления	14
Текстовые сообщения	14
Элементы управления в виде текстовых окон	14
Управляющие командные кнопки	15
Управляющие кнопки тумблеров	16
Элементы управления LED	16
Управляющие двоичные переключатели	17
Элементы управление набором	17
Элементы управления окном списка	19
Элементы оформления	20
Графические элементы управления	20
Элементы управления ленточными схемами	22
Элементы управления рисунками	22
Таймеры	23
Использование всплывающих панелей	23
Работа с всплывающей панелью сообщений	24
Работа с всплывающей панелью обобщенных сообщений	24
Работа с всплывающей панелью приглашений	24
Работа с всплывающей панелью подтверждения	24
Работа с всплывающей панелью выбора файла	25
Работа с графической всплывающей панелью	25
Использование шрифтов	26
Меташрифты, использующие шрифты, типичные для каждой платформы	26
Шрифты, использующие типы шрифтов, типичные для каждой платформы	26
Меташрифты, использующие типы шрифтов, инсталлированные LabWindows/CVI	26

Общая информация о Редакторе пользовательского интерфейса	27
Использование всплывающих меню Редактора пользовательского интерфейса	28
Общая информация о Построителе кодов	28
Меню Редактора пользовательского интерфейса	29
Меню File	29
Команды New, Open, Save и Exit LabWindows/CVI.....	29
Команды Save As и Close.....	29
Save Copy As	29
Save All	30
Add File to Project	30
Read Only.....	30
Print	30
Меню Edit.....	30
Undo и Redo	31
Cut и Copy	31
Paste	31
Delete	31
Copy Panel и Cut Panel.....	32
Menu Bars	32
Panel.....	34
Control.....	36
Порядок табуляции	38
Set Default Font	39
Apply Default Font.....	39
Control Style	39
Меню Create	40
Panel.....	40
Menu Bar.....	40
Элементы управления.....	40
Меню View	41
Find UIR Objects... ..	41
Show/Hide Panels	42
Preview User Interface Header File	43
Меню Arrange	43
Alignment.....	43
Align Horizontal Centers	44
Distribution... ..	44
Distribute Vertical Centers.....	45
Control ZPlane Order	45
Center Label	45
Control Coordinates... ..	45
Меню Code	46
Set Target File.....	46
Generate	47
All Code... ..	48
Main Function.....	49
All Callbacks.....	50
Panel Callback.....	50
Control Callbacks.....	50
Menu Callbacks.....	50
View	51
Preferences	52

Default Panet Events... и Default Control Events.....	52
Always Append Code to End.....	52
Меню Run.....	53
Меню Library	53
Меню Window.....	53
Меню Options.....	53
Operate Visible Panels.....	53
Next Tool	54
Preferences.....	54
Undo Preferences.....	54
Color Preferences.....	55
Constant Name Assignment.....	55
Assign Missing Constants.....	55
Save in Text Format.....	55
Load From Text Format.....	56

Информация о данном руководстве

В *Справочном руководстве по пользовательскому интерфейсу LabWindows/CVI* содержится информация о создании и управлении заказными пользовательскими интерфейсами с помощью Библиотеки пользовательского интерфейса LabWindows/CVI. Чтобы эффективно пользоваться данным руководством, необходимо ознакомиться с информацией, представленной в *Запуске с помощью LabWindows/CVI* и *Справочном руководстве пользователя LabWindows/CVI*. Кроме того, необходимо знать, как выполняются базовые задачи с помощью LabWindows/CVI и Windows.

Организация справочного руководства

Справочное руководство по пользовательскому интерфейсу LabWindows/CVI организовано следующим образом:

- Глава 1, *Концепции пользовательского интерфейса*, описывает построение и управление графическим пользовательским интерфейсом в LabWindows/CVI. Дано объяснение, как взаимодействуют ресурсные файлы (.uir) вашего пользовательского интерфейса с файлами кодов, что поможет Вам правильно построить свои программы, запускаемые событием. Кроме того, в главе 1 представлено описание объектов и концепций, относящихся к Графическому пользовательскому интерфейсу (GUI). А также описано, как с помощью мышки и клавиатуры работать со строками меню, панелями и элементами управления.
- Глава 2, *Справочник редактора пользовательского интерфейса*, описывает, как интерактивно создавать GUI. Представлено описание Редактора пользовательского интерфейса и процедур по созданию и редактированию панелей, элементов управления и строк меню.

Условные обозначения, использованные в настоящем руководстве

В руководстве использованы следующие условные обозначения.

bold	Текст, выделенный полужирным шрифтом, означает параметр, элемент меню, возвращаемое значение, элемент панели функций, или кнопку или опцию диалогового окна.
<i>italic</i>	Текст, выделенный курсивом, используется для привлечения особого внимания, указания на перекрестную ссылку или на ключевую концепцию.
<i>bold italic</i>	Текст, выделенный одновременно полужирным шрифтом и курсивом, используется для предупреждающих или требующих особого внимания сообщений.
monospace	Текст с использованием данного шрифта означает текст или символы, которые Вы должны вводить с клавиатуры буквально. Этим же шрифтом приведены разделы кодов, примеры программирования и примеры синтаксиса. Данный шрифт используется для собственных имен дисководов, путей, директорий, программ, подпрограмм, подмаршрутов, имен устройств, переменных, имен файлов и расширений, а также для операторов и комментариев, взятых из программного кода.
<i>italic monospace</i>	Слова с использованием данного шрифта, выделенные курсивом, означают, что Вы должны подставить соответствующие слова или значения вместо выделенных слов.
< >	В угловые скобки заключаются названия клавиш. Знак тире между двумя или более названиями клавиш, заключенных в угловые скобки, означает, что Вы должны одновременно нажать указанные клавиши, например, <Ctrl-Alt-Delete>.
»	Символ » проводит Вас через встроенные элементы меню и опции диалогового окна к конечному действию. Последовательность File » Page Setup » Options » Substitute Fonts означает, что сначала надо развернуть меню File , затем выбрать элемент Page Setup , выбрать Options и, наконец, выбрать из последнего диалогового окна опцию Substitute Fonts .
paths	В настоящем справочнике пути (paths) обозначаются с использованием обратной наклонной черты (\), разделяющей имена дисководов, директории и файлы, как показано в данном примере. <code>drivename\dir1name\dir2name\myfile.</code>

Связь с заказчиком

National Instruments будет рада получить Ваши комментарии о наших продуктах и документации. Нас интересует, какие приложения Вы разрабатываете с помощью наших продуктов, и мы всегда будем рады помочь Вам при возникновении каких-либо проблем.

Глава 1. Концепции пользовательского интерфейса

В настоящей главе описано построение и управление графическим пользовательским интерфейсом в LabWindows/CVI. Дано объяснение, как взаимодействуют ресурсные файлы (.uir) Вашего пользовательского интерфейса с файлами кодов, что поможет Вам правильно построить свои программы, запускаемые событием. Кроме того, в главе представлено описание объектов и концепций, относящихся к Графическому пользовательскому интерфейсу (GUI). А также описано, как с помощью мышки и клавиатуры работать со строками меню, панелями и элементами управления.

Введение в Графический пользовательский интерфейс

Графический пользовательский интерфейс LabWindows/CVI (GUI) может состоять из панелей, командных кнопок, выпадающих меню, графиков, схем, ручек управления, счетчиков и многих других элементов управления и индикаторов. На Рисунке 1-1 представлен типичный GUI, созданный с помощью LabWindows/CVI. В LabWindows/CVI Вы можете создавать GUI интерактивно, используя Редактор пользовательского интерфейса, перемещающий редактор с инструментами для конструирования, организовывая нужным Вам образом элементы заказного пользовательского интерфейса. С помощью интерактивного Редактора пользовательского интерфейса Вы можете создать обширный GUI для своей программы, не написав ни одной строки кодов. Закончив создание GUI в Редакторе пользовательского интерфейса, Вы сохраняете GUI как файл Ресурсов пользовательского интерфейса (.uir). В Главе 2, *Справочник по Редактору пользовательского интерфейса*, представлена детальная информация о Редакторе пользовательского интерфейса.

В дополнение к Редактору пользовательского интерфейса, Библиотека пользовательского интерфейса также имеет функции по оперативному созданию или изменению элементов управления в Вашем GUI, так что Вы можете программно дополнять, изменять или создавать весь GUI.

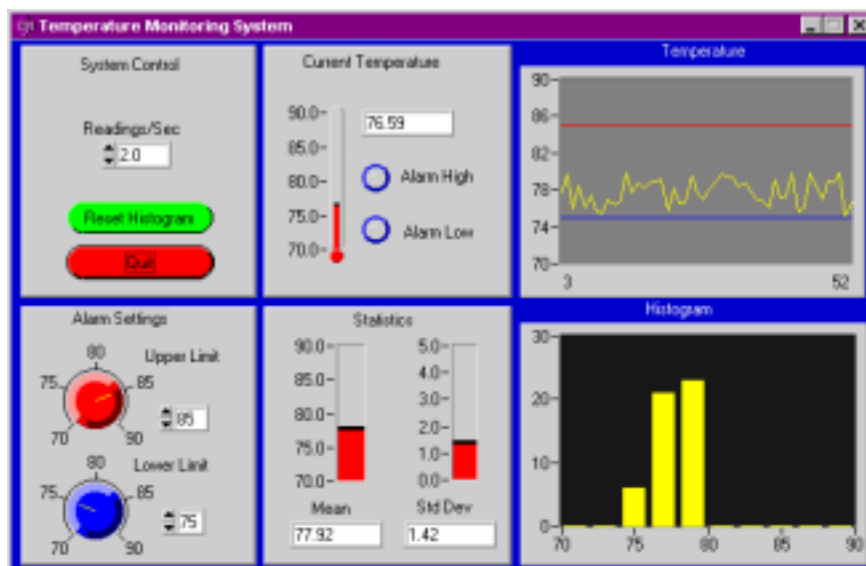


Рисунок 1-1. Типичный Графический пользовательский интерфейс LabWindows/CVI.

События пользовательского интерфейса

При разработке пользовательского интерфейса Вы определяете области на экране компьютера в форме элементов управления, которые могут генерировать события. Например, если Вы нажимаете командную кнопку, то эта кнопка генерирует событие пользовательского интерфейса, которое затем передается в Вашу программу C. Фактически, элементы управления LabWindows/CVI генерируют множество событий пользовательского интерфейса. Например, с помощью одного нажатия мышкой на командную кнопку в Вашу программу будут переданы для обработки следующие события пользовательского интерфейса:

1. Событие GOT_FOCUS - Если командная кнопка не является активным элементом управления (не имеет фокуса ввода), то одно нажатие мышкой на данную кнопку сделает ее активным элементом управления. Когда элемент управления получает фокус ввода, происходит событие GOT_FOCUS.
2. Событие LEFT_CLICK - Если пользователь нажимает левой кнопкой мышки на командную кнопку, происходит событие LEFT_CLICK. Элементы управления пользовательского интерфейса LabWindows/CVI могут распознавать нажатие левой, правой, одной и обеих кнопок мышки одновременно.
3. Событие COMMIT - Когда пользователь отпускает кнопку мышки, происходит событие COMMIT, означающее, что пользователь выполнил с помощью элемента управления событие commit.

Различные типы элементов управления, имеющиеся в Редакторе пользовательского интерфейса LabWindows/CVI, выдают различные типы информации и передают различные события пользовательского интерфейса в Вашу программу. В первой половине данной главы представлен каждый тип элементов управления, имеющихся в LabWindows/CVI, а также дана информация о том, как использовать каждый из этих типов и какие события могут передаваться с помощью каждого типа элементов управления.

Управление пользовательским интерфейсом с помощью обратных вызовов или GetUserEvent

После создания собственного GUI в LabWindows/CVI Вы должны разработать программу C для обработки событий, генерированных пользовательским интерфейсом, и управления выполнением Вашей программы. LabWindows/CVI предлагает два базовых метода разработки программ: с помощью функций обратных вызовов и с помощью циклов событий. Если используются функции обратного вызова, то Вы пишете в свою программу отдельные функции, которые вызываются непосредственно элементами управления пользовательским интерфейсом. Например, в Вашей программе может быть функция AcquireData, которая присваивается кнопке **Acquire**. После этого, когда бы не нажималась кнопка **Acquire**, вся информация, генерированная этой кнопкой, будет передаваться непосредственно функции AcquireData, где она будет обрабатываться.

По второму методу Вы можете использовать цикл событий для обработки событий своего пользовательского интерфейса в LabWindows/CVI. При использовании цикла событий все происходящие события проходят вызов к функции GetUserEvent в Вашей программе. Функция GetUserEvent определяет, какие элементы управления генерировали данные события, и Ваша программа соответственно их обрабатывает.

В зависимости от Вашей программы, как функции обратного вызова, так и цикл событий имеют определенные преимущества. В программе Вы можете использовать любой из данных методов или комбинацию двух методов, что обеспечит дополнительную гибкость.

Связь с помощью исходного кода

Для установления связи между Вашими файлами `.uir` и исходными файлами C Вы должны включить в свой исходный код файл-заголовок. При создании элемента управления в Редакторе Пользовательского интерфейса Вы должны задать для элемента управления определенную информацию так, чтобы Ваша программа могла корректно получать к нему доступ. Каждому элементу управления задается *постоянное имя* (*constant name*) в Редакторе Пользовательского интерфейса. Постоянное имя используется в функциях Библиотеки Пользовательского интерфейса Вашей программы для определения элементов управления в Вашем GUI. Например, Вы определяете цифровой вывод на своем пользовательском интерфейсе с помощью постоянного имени `READOUT`, находящегося на панели с постоянным именем `PANEL`. Для установки значения, выданного в выводе Вашей программы, Вы вызвали бы функцию `SetCtrlVal` с именем `PANEL_READOUT` как один из параметров при задании элемента управления, в котором должно быть установлено данное значение.

Дополнительно к присвоению постоянных имен каждому элементу управления Вашего GUI, Вы также можете присвоить функции обратного вызова элементам управления в Редакторе Пользовательского интерфейса. Например, если Вы хотите, чтобы при каждом нажатии командной кнопки `ACQUIRE` выполнялась функция `AcquireData`, то Вам необходимо присвоить кнопке `ACQUIRE` имя функции обратного вызова `AcquireData` в Редакторе Пользовательского интерфейса.

Если Вы сохраните файл `.uir` в Редакторе Пользовательского интерфейса, LabWindows/CVI автоматически создаст соответствующий файл-заголовок (`.h`) с тем же базовым именем файла. Файл-заголовок объявляет все постоянные имена и функции обратного вызова, присвоенные Вами в рамках файла `.uir`. Включение файла-заголовка в Ваш исходный код приведет к тому, что все функции обратной связи и постоянные имена, связанные с Вашим GUI, будут объявляться автоматически.

Управляющие режимы для генерации событий `Commit` (исполнения)

События `commit` (исполнения) генерируются, когда пользователь GUI фактически приступает к выполнению такой операции, как выбор из меню или ввод номера, и нажимает `<Enter>`. Когда Вы создаете элемент управления, Вы можете присвоить ему один из следующих режимов, определяющих, как данный элемент управления будет генерировать события `commit`.

- Нормальный
- Индикатор
- Горячий
- Проверка достоверности

Нормальный режим означает, что пользователь может работать с элементом управления и что данный элемент может быть изменен программно.

Индикатор показывает, что Вы можете изменить элемент управления программно, но пользователи не могут генерировать события `commit` или события с измененным значением. Элементы управления для схем или текстовых сообщений всегда имеют режим индикатора.

Горячий режим идентичен нормальному, за исключением того, что элемент управления генерирует событие `commit`, когда он активизируется пользователем. Обычно, элемент управления в горячем режиме генерирует событие `commit`, когда изменяется его состояние. Например, когда пользователь устанавливает двоичный переключатель в состояние “включено”, начинает генерироваться событие `commit`.

Приведенные ниже типы элементов управления имеют специфические правила относительно того, как генерируются события `commit`.

- Для горячих элементов управления текстовыми окнами, а также цифровых и строчных, событие `commit` генерируется, когда пользователь нажимает `<Enter>` или `<Tab>` или нажимает мышкой в любом месте после ввода значения в элемент управления.
- Для горячих элементов управления списком выделенных участков, не находящихся в режиме проверки, событие `commit` генерируется, когда пользователь нажимает клавишу `<Enter>` при активном элементе управления или дважды нажимает на элемент списка.
- Для горячих элементов управления списком выделенных участков, находящихся в режиме проверки, событие `commit` генерируется, когда пользователь нажимает клавишу пробела при активном элементе управления или дважды нажимает на элемент списка.
- Для горячих элементов управления графиками событие `commit` генерируется, когда пользователь перемещает курсор с помощью клавиши со стрелкой, или когда пользователь отпускает кнопку мышки после перемещения курсора.

Проверка достоверности является режимом, идентичным горячему, за исключением того, что перед генерацией события `commit` программа проверяет достоверность всех цифровых элементов управления на панели, атрибут проверки диапазона которых установлен на `Notify` (уведомить). LabWindows/CVI проверяет значение элемента управления относительно заданного диапазона. Если обнаруживается недостоверное условие, LabWindows/CVI активизирует элемент управления, вызывая появление окна уведомления, подобного приведенному на Рисунке 1-2. Элемент управления в режиме проверки достоверности не сможет генерировать событие `commit` до тех пор, пока пользователь не введет новое достоверное значение во все элементы управления, оказавшиеся вне указанного диапазона. Данная процедура обеспечивает достоверность всех цифровых/скалярных элементов управления на момент сообщения о событии `commit` Вашей прикладной программе.

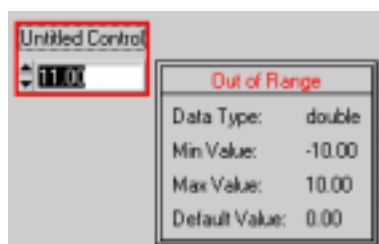


Рисунок 1-2. Диалоговое окно Out of Range (вне диапазона).

Использование CodeBuilder (Построителя кода) для создания исходного кода Вашего GUI

С помощью LabWindows/CVI CodeBuilder Вы можете автоматически создавать полный код C, который компилируется и выполняется на основе создаваемого или редактируемого Вами файла пользовательского интерфейса (`.uir`). Выбирая определенные опции, имеющиеся в меню **Code** Редактора Пользовательского интерфейса, Вы можете создать *план программы (skeleton code)*. План программы синтаксически и программно корректирует код, который компилируется и выполняется до того, как Вы введете хотя бы одну строку кода. Благодаря свойствам CodeBuilder, Вы сберегаете время на печать в стандартном коде, входящем в каждую программу, исключаете синтаксические ошибки и ошибки печати и поддерживаете организацию файла исходного кода в соответствии со стилем программирования. Так как программа CodeBuilder компилируется и выполняется одновременно, Вы можете разрабатывать и испытывать свой проект, работая над каждой функцией по очереди.

Работа Графического пользовательского интерфейса

Перед созданием первого GUI с использованием LabWindows/CVI необходимо четко понять, какие графические объекты существуют и как они работают. С помощью настоящего раздела Вы сможете изучить работу всех элементов управления и индикаторов GUI.

Использование панелей

Панель представляет собой четырехугольную область экрана, в которой находятся элементы управления и строки меню. Панели обеспечивают поддержку различных функций, как, например, представление инструмента передней панели.

Панели действуют подобно окнам. Вы можете уменьшить панели, изменить их форму, переместить их, наложить несколько панелей друг на друга или вложить одну в другую. На Рисунке 1-3 представлен пример дочерней панели, вложенной в родительскую. Внешняя панель называется родительской. Дочерняя панель может вкладываться не только в родительскую, но и в другую дочернюю панель. Вы не можете вытащить дочернюю панель из ее родительской панели. Если родительская панель имеет сжатый вид, дочерняя панель может быть полностью или частично скрыта.

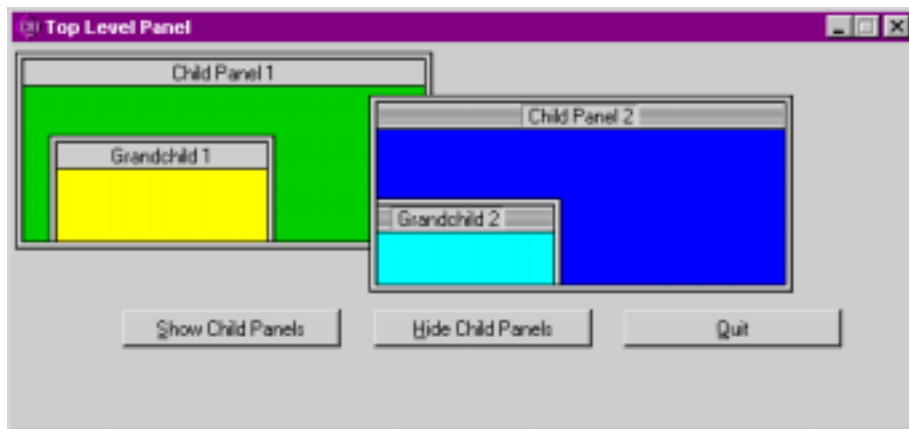


Рисунок 1-3. Расположение дочерней панели в родительской.

Для работы с панелями с клавиатуры:

- Нажмите <Shift-Ctrl> и клавишу со стрелкой, направленной вверх, чтобы переместиться в родительскую панель текущей панели.
- Нажмите <Shift-Ctrl> и клавишу со стрелкой, направленной вниз, чтобы переместиться в дочернюю панель текущей панели.
- Нажмите <Shift-Ctrl> и клавишу со стрелкой, направленной влево или вправо, для перемещения с одной панели на другую панель того же уровня в иерархии панелей.
- Нажмите <Tab> для перемещения на следующий элемент управления в панели.
- Нажмите <Shift-Tab> для перемещения на предыдущий элемент управления в панели.

Для работы с панелью с помощью мышки:

- Для активизации панели нажмите мышкой в любом месте на панели.
- Для активизации элемента управления нажмите мышкой в любом месте на этом элементе управления.

Использование линеек меню

Линейка меню представляет собой механизм, заключающий в себя набор команд. Линейка меню появляется в виде линейки в верхней части панели и содержит набор заголовков меню. На Рисунке 1-4 представлена простая линейка меню с выпадающим меню.

Test	Instruments	Options	Quit!
	Voltmeter Scope Counter		

Рисунок 1-4. Линейка меню и выпадающее меню.

Заголовок меню, заканчивающийся на восклицательный знак, называется *меню немедленного действия* и не снабжен выпадающим меню. Если пользователь нажимает мышкой на меню немедленного действия, то оно выполняется сразу же.

Заголовок меню без восклицательного знака снабжен выпадающим меню. Оно содержит набор элементов меню и заголовков подменю. Элементы подменю выдаются сбоку от заголовков подменю. На Рисунке 1-5 приведен пример подменю.

Test	Instruments	Options	Quit!
	Voltmeter Scope Counter		
		HP 54501A Tek 2430A	

Рисунок 1-5. Выпадающее меню с подменю.

Работа с линейкой меню с клавиатуры:

- Для выдачи меню на экран или выполнения меню немедленного действия нажмите клавишу <Alt> и подчеркнутую букву заголовка меню. Меню появляется на экране с высвеченным заголовком.
- Для выдачи на экран прилегающих меню нажмите клавишу со стрелкой, направленной вправо или влево.
- Для выбора элемента в меню нажмите клавишу со стрелкой, направленной вверх или вниз, или подчеркнутую букву элемента меню. Выбранный элемент меню появится на экране с высвеченной меткой.
- Для выполнения высвеченного элемента меню и удаления меню с экрана нажмите <Enter>.
- Для удаления меню с экрана без выполнения какого-либо его элемента нажмите <Esc>.

Работа с линейкой меню с помощью мышки:

- Для выдачи меню на экран или выполнения меню немедленного действия нажмите мышкой на заголовок меню.
- Для выполнения элемента меню нажмите мышкой на этот элемент. Меню исчезнет.
- Для удаления меню с экрана без выполнения какого-либо его элемента нажмите мышкой в любом месте экрана вне меню.

Неяркие заголовки или элементы меню в данный момент не активны (не могут быть выбраны).

Использование элементов управления

Элемент управления представляет собой расположенный на панели объект, принимающий вводы пользователя и выдающий информацию. Библиотека Пользовательского интерфейса поддерживает следующие типы элементов управления:

- Цифровые
- Строчные
- Текстовое сообщение
- Текстовое окно
- Кнопку команды
- Кнопку тумблера
- LED
- Двоичный переключатель
- Элементы управление набором
- Окно списка
- Элементы оформление
- Графики и схемы
- Рисунки
- Таймер

Чтобы элемент управления был готов принять ввод, нажмите мышкой на элемент управления или нажмите <Tab> или <Shift-Tab> для перемещения с одного элемента управления на другой. Нажатие <Alt> и подчеркнутой буквы в метке элемента управления приводит элемент управления в готовность принять ввод при условии, что строки меню недоступны.

Типы данных элементов управления

Каждый элемент управления имеет связанный с ним *тип данных*. Тип данных элемента управления определяет тип данных переменных, используемых для установки и получения значения элемента управления. В приведенном ниже списке представлены типы данных элементов управления LabWindows/CVI.

```
unsigned char
char
char *
unsigned short int
short int
unsigned long int
long int
float
double
```

Не все типы данных достоверны для каждого типа элементов управления.

Цифровые элементы управления

Цифровые элементы управления используются для ввода или дисплея цифровых значений. Одним из типичных примеров их использования может служить ввод или дисплей величины напряжения. Цифровые элементы управления могут иметь различные графические изображения, включая кнопки, счетчики, термометры, измерительные приборы и наборные диски. На Рисунке 1-6 приведены цифровые элементы управления.

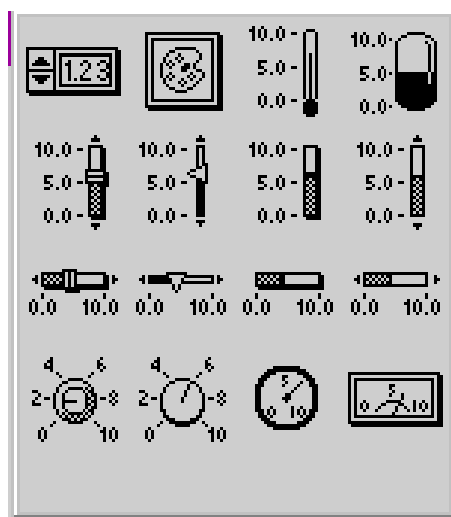


Рисунок 1-6. Цифровые элементы управления.

Работа с цифровыми элементами управления с клавиатуры:

- Для перемещения курсора в пределах цифрового дисплея элемента управления нажать клавишу со стрелкой, направленной вправо или влево.
- Для увеличения или уменьшения значения, выданного в элементе управления, нажать клавишу со стрелкой, направленной вверх или вниз.
- Для перемещения курсора в начало текста нажать <Home>.
- Для перемещения курсора в конец текста нажать <End>.
- Для высвечивания всего текста слева от курсора нажать <Shift-Home>.
- Для высвечивания всего текста справа от курсора нажать <Shift-End>.

Работа с цифровым элементом управления с использованием мышки:

- Для изменения значения элемента управления нажать на цифровом дисплее мышкой стрелки вверх/вниз.
- Для изменения значения элемента управления нажать и вытянуть иглу (для круглых элементов управления, например, кнопок) или слайдер (для плавных элементов управления).
- Для установки элемента управления на значение метки нажать мышкой на метки шкалы.

Если цифровой элемент управления находится в горячем режиме или режиме проверки достоверности, событие `commit` генерируется, когда пользователь нажимает клавиши со стрелками, направленными вверх или вниз, или когда он изменяет цифровой дисплей цифрового элемента управления, а затем нажимает клавишу <Enter> или <Tab> или нажимает мышкой на другой объект.

Строчные элементы управления

Строчные элементы управления используются для ввода или просмотра строк символьных значений. Вы можете использовать эти элементы управления для ввода имени лица. На Рисунке 1-7 приведен строчный элемент управления.

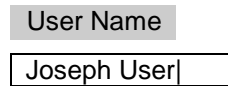


Рисунок 1-7. Строчный элемент управления.

Работы со строчным элементом управления:

- Для перемещения курсора в пределах строчного элемента управления нажмите клавишу со стрелкой, направленной вправо или влево.
- Для перемещения курсора в начало текста нажмите <Home>.
- Для перемещения курсора в конец текста нажмите <End>.

Если режим элемента управления строки соответствует горячему или проверке достоверности, то событие commit генерируется, когда пользователь изменяет строку, а затем нажимает клавишу <Enter> или <Tab> или нажимает мышкой на другой объект.

Текстовые сообщения

Текстовые сообщения представляют собой индикаторы, используемые для программной выдачи строк текста. Вы не можете работать с ними с клавиатуры, но можете задать для них функции обратного вызова так, чтобы они реагировали на события, вызываемые нажатием кнопки мышки. На Рисунке 1-8 представлен элемент управления в виде текстового сообщения.

Warning: Temperature Too High!

Рисунок 1-8. Текстовое сообщение.

Элементы управления в виде текстовых окон

Текстовые окна представляют собой большие строчные элементы управления, характеризующиеся режимами заворачивания строк, слов и символов и прокручивающие строки, что позволяет выдавать большие объемы текста. Как и при строчных элементах управления, Вы можете интерактивно вводить текст в текстовые окна и управлять текстовым окном программно.

На Рисунке 1-9 приведен пример текстового окна.

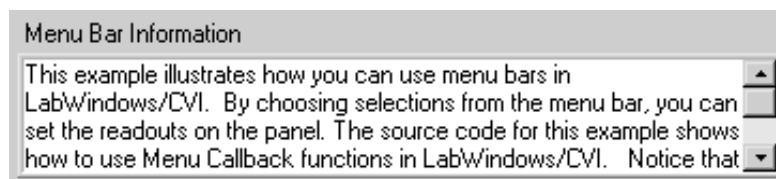


Рисунок 1-9. Управление текстовым окном.

Для ввода текста в текстовое окно:

- Для вставки четырех пробелов нажмите <Ctrl-Tab>.
- Для начала новой строки нажмите <Ctrl-Enter>.
- Для перехода в начало или конец текущей строки нажмите <Home> или <End>.
- Для прокрутки текстового окна на страницу назад или вперед нажмите <Page Up> или <Page Down>.
- Отрывок текста может быть выбран, если нажать клавишу <Shift> и не отпуская ее нажимать клавиши <Home>, <End>, <Page Up> или <Page Down>.

Если элемент управления текстовым окном находится в горячем режиме или режиме проверки достоверности, событие commit будет генерироваться, если пользователь изменит значение текстового окна, а затем нажмет клавишу <Enter> или <Tab> или нажмет мышкой на другой объект.

Управляющие командные кнопки

Командные кнопки используются для запуска действия, указанного на метке кнопки. На Рисунке 1-10 показаны командные кнопки.



Рисунок 1-10. Командные кнопки.

Для управления командной кнопкой с клавиатуры: нажмите <Enter> или <space>, чтобы активизировать кнопку. Если кнопка находится в горячем режиме управления или в режиме оценки достоверности, событие commit начнет генерироваться после нажатия пользователем клавиши <Enter> или <space>. На мгновение внешний вид кнопки изменится, показывая, что выбрана была именно эта кнопка.

Для управления кнопкой запуска с помощью мышки нажмите мышкой на данную кнопку. Кнопка будет находиться в нажатом положении до тех пор, пока пользователь не отпустит клавишу мышки или не уберет мышку с кнопки. Если командная кнопка находится в горячем режиме управления или в режиме проверки достоверности, событие commit начнет генерироваться, когда пользователь нажмет кнопкой мышки на область командной кнопки, а затем отпустит ее. Если пользователь отождмет клавишу мышки вне области командной кнопки, событие commit генерироваться не будет.

Управляющие кнопки тумблеров

Кнопки тумблеров позволяют сделать выбор между двумя различными состояниями. Кнопка тумблера имеет два положения: нажатое и отжатое. Когда кнопка нажата, ее значение соответствует 1, а когда она отжата, значение соответствует 0. На Рисунке 1-11 представлены кнопки тумблеров.

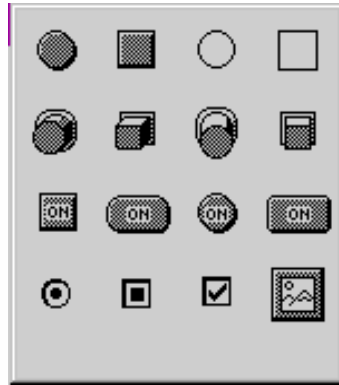


Рисунок 1-11. Управляющие кнопки тумблеров.

Для работы с кнопками тумблеров с клавиатуры:

- Для приведения кнопки в нажатое положение нажмите <Home> или клавишу со стрелкой, направленной вверх.
- Для приведения кнопки в отжатое положение нажмите <End> или клавишу со стрелкой, направленной вниз.
- Для изменения состояния кнопки нажмите <space> или <Enter>.

При управлении кнопкой тумблера с помощью мышки нажмите на нее мышкой для изменения состояния кнопки.

Если кнопка тумблера находится в горячем режиме управления или в режиме проверки достоверности, событие commit начнет генерироваться, когда кнопка активна, и пользователь изменит ее состояние.

Элементы управления LED

LED (светодиоды) указывают на состояние включено/выключено. Когда LED включен, его значение соответствует 1, а его цвет соответствует состоянию ON (включено). Когда LED отключен, его значение соответствует 0, а его цвет на экране соответствует состоянию OFF (отключено). На Рисунке 1-12 приведены элементы управления LED.



Рисунок 1-12. Элементы управления LED.

Светодиоды работают так же, как и управляющие кнопки тумблеров.

Управляющие двоичные переключатели

Двоичные переключатели, как и кнопки тумблеров, позволяют сделать выбор между двумя состояниями: включено и выключено. Вы также можете привязать определенное значение к каждому состоянию двоичного переключателя. На Рисунке 1-13 приведены двоичные переключатели.



Рисунок 1-13. Управляющие двоичные переключатели.

Управляющие двоичные переключатели работают так же, как и управляющие кнопки тумблеров.

Элементы управление набором

Элементы управления набором используются для выбора из группы элементов. Многие элементы управления набором похожи на цифровые элементы управления, но они представляют собой законченный набор пар метка/значение. На Рисунке 1-14 представлены элементы управления набором.

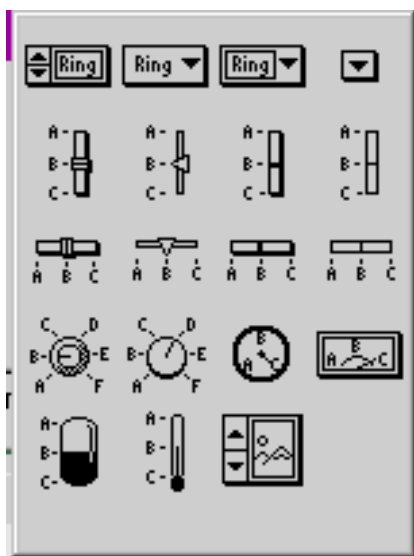


Рисунок 1-14. Элементы управления набором.

Для работы с элементами управления набором с клавиатуры:

- Для выбора предыдущего элемента нажмите клавишу со стрелкой, направленной вверх.
- Для выбора следующего элемента нажмите клавишу со стрелкой, направленной вниз.

Элементы управления набором со стрелкой могут быть также в всплывающем формате, в котором на экране одновременно появляется линейный список всех элементов управления набором. На Рисунке 1-15 приведен пример элементов управления набором в всплывающем формате.

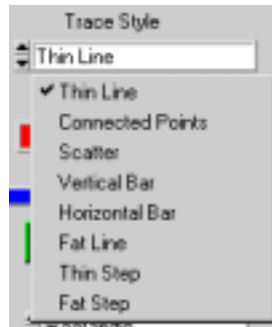


Рисунок 1-15. Элементы управления набором в всплывающем формате

Для работы с элементами управления набором в всплывающем формате с клавиатуры:

- Для выдачи на дисплей всплывающих элементов управления нажать <space>.
- Для высвечивания определенного внутреннего элемента нажать клавиши со стрелками, направленными вверх или вниз.
- Для выбора высвеченного элемента нажать <Enter>. Всплывающий элемент исчезнет и появится только что выбранный новый элемент управления.
- Для отключения всплывающих элементов управления без изменения выбранного элемента нажать <Esc>.

Работа внутри элемента управления с помощью мышки:

- Если элемент управления имеет стрелки, нажмите на стрелки, чтобы сделать выбор внутри этого элемента.

Работа с элементами управления набором с помощью мышки:

- Для выдачи на дисплей всплывающего меню элемента управления нажмите на элемент управления мышкой.
- Для выбора внутреннего элемента нажмите на него мышкой. Всплывающий элемент управления исчезнет и появится только что выбранный новый элемент управления.
- Если всплывающие элементы превышают размер экрана, нажмите кнопкой мышки на верхний элемент и держите ее в таком положении для прокрутки вверх.
- Если всплывающие элементы превышают размер экрана, нажмите кнопкой мышки на нижний элемент и держите ее в таком положении для прокрутки вниз.
- Для завершения операции и удаления всплывающих элементов управления нажмите кнопкой мышки в любом месте вне меню.

Внимание: Если всплывающие элементы управления набором в два раза превышают размер экрана, пользователи используют вместо них окно списка. Смотрите раздел *Элементы управления окном списка для инструкций по работе с элементами управления окном списка.*

Если элементы управления набором находятся в горячем режиме или в режиме проверки достоверности, событие commit начинает генерироваться, когда пользователь изменяет значение элемента управления.

Элементы управления окном списка

Элементы управления окном списка используются для выбора элемента из списка. На Рисунке 1-16 приведен пример управления выбором из списка.

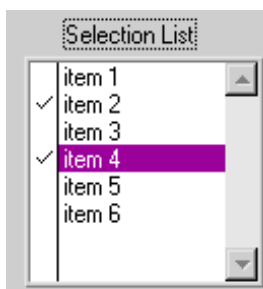


Рисунок 1-16. Управление выбором из списка в режиме проверки.

С помощью линейки прокрутки, расположенной справа от списка, можно прокручивать список вперед и назад.

Работа с окном списка с клавиатуры:

- Для высвечивания предыдущего элемента списка нажать клавишу со стрелкой вверх.
- Для высвечивания следующего элемента списка нажать клавишу со стрелкой, направленной вниз.
- Для перехода к первому элементу списка нажать <Home>.
- Для перехода к концу списка нажать <End>.
- Для прокрутки на одну страницу назад нажать <Page Up>.
- Для прокрутки на одну страницу вперед нажать <Page Down>.
- Для нахождения какого-либо элемента в списке используйте функцию Quick Type (Быстрая печать). Если Вы введете А, Quick Type найдет первый случай появления элемента списка, начинающегося с “А”. Если Вы введете полное имя элемента, Quick Type перейдет к данному элементу. (При нажатии клавиши с задержкой не менее одной секунды буфер Quick Type очищается.)
- Для перехода к предыдущему элементу в буфере Quick Type нажмите клавишу со стрелкой, направленной влево.
- Для перехода к следующему элементу в буфере Quick Type нажмите клавишу со стрелкой вправо.
- Если окно списка находится в режиме проверки, то для переключения метки проверки на текущем элементе нажмите <space> или <Enter>.

Работа с управлением окном списка с помощью мышки:

- Для высвечивания элемента списка нажмите на него кнопкой мышки; если Вы находитесь в режиме проверки, данная операция переключит метку проверки.
- Для выбора элемента дважды нажмите на него кнопкой мышки. Вы не должны находиться в режиме проверки.
- Для прокрутки списка нажмите, не отпуская, кнопкой мышки на стрелку.
- Нажмите кнопкой мышки на кнопку линейки прокрутки и передвиньте ее.
- Для прокрутки на одну страницу назад нажмите кнопкой мышки над кнопкой на линейке прокрутки.
- Для прокрутки на одну страницу вперед нажмите кнопкой мышки под кнопкой на линейке прокрутки.

Если окно списка находится в горячем режиме или режиме проверки достоверности, то для генерации события commit:

- если элемент управления установлен на **Check Mode (режим проверки)**, выберите необходимый элемент в окне списка и нажмите <space> или нажмите на него кнопкой мышки.
- если элемент управления не установлен на **Check Mode (режим проверки)**, выберите необходимый элемент в окне списка и нажмите <space> или дважды нажмите на него кнопкой мышки.

Элементы оформления

Элементы оформления используются для усиления визуальной стороны GUI. Они не содержат данных, но элементам оформления могут быть приписаны функции обратного вызова, так что они смогут реагировать на события, вызываемые нажатием кнопки мышки. На Рисунке 1-17 представлены элементы оформления.

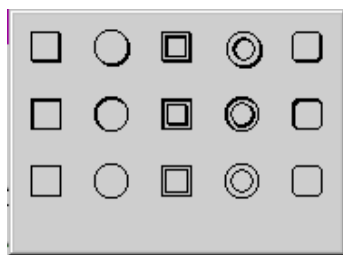


Рисунок 1-17. Элементы оформления GUI.

Графические элементы управления

Графические элементы управления выдают графические данные в виде одного или более графиков. *График* состоит из кривой, точки, геометрической формы или текстовой строки. На Рисунке 1-18 приведен пример графического элемента управления.

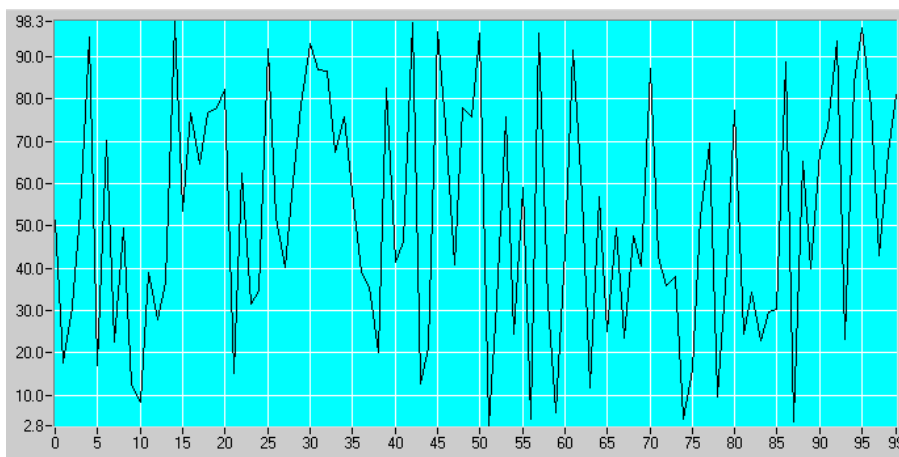


Рисунок 1-18. Графический элемент управления.

Графические элементы управления могут иметь один или более связанных с ними курсоров. С помощью курсора Вы можете выбрать точку или область на графике для дальнейшей их обработки или анализа. Если графический элемент управления находится в горячем режиме, курсоры будут генерировать события commit.

Работа с графиком с помощью курсоров с клавиатуры: используйте клавиши из приведенных ниже таблиц как для курсоров произвольного формата, так и для курсоров в формате захвата точки.

Таблица 1-1. Клавиши для выбора курсоров

Когда Вы нажимаете...	Вы выбираете...
<Page-Up>	Предыдущий курсор.
<Page-Down>	Следующий курсор.
Клавишу со стрелкой влево	Левые десять точек растра.
Клавишу со стрелкой вправо	Правые десять точек растра.
Клавишу со стрелкой вверх	Верхние десять точек растра.
Клавишу со стрелкой вниз	Нижние десять точек растра.
<Shift>-клавишу со стрелкой влево	Одну точку растра слева.
<Shift>-клавишу со стрелкой вправо	Одну точку растра справа.
<Shift>-клавишу со стрелкой вверх	Одну верхнюю точку растра.
<Shift>-клавишу со стрелкой вниз	Одну нижнюю точку растра.
<Ctrl>-клавишу со стрелкой влево	Переход к левому краю области графика.
<Ctrl>-клавишу со стрелкой вправо	Переход к правому краю области графика.
<Ctrl>-клавишу со стрелкой вверх	Переход к верхнему краю области графика.
<Ctrl>-клавишу со стрелкой вниз	Переход к нижнему краю области графика.
<Home>	Переход к нижнему левому углу области графика.
<End>	Переход к верхнему правому углу области графика.
Клавишу со стрелкой влево	Предыдущую точку текущего графика.
Клавишу со стрелкой вправо	Следующую точку текущего графика.
Клавишу со стрелкой вверх	Следующую точку текущего графика.
Клавишу со стрелкой вниз	Предыдущую точку текущего графика.
<Shift>-клавишу со стрелкой влево	На 10 точек назад на текущем графике.
<Shift>-клавишу со стрелкой вправо	На 10 точек вперед на текущем графике.
<Shift>-клавишу со стрелкой вверх	На 10 точек вперед на текущем графике.
<Shift>-клавишу со стрелкой вниз	На 10 точек назад на текущем графике.
<Ctrl>-клавишу со стрелкой влево	Ближайшую точку слева в направлении X на текущем графике.
<Ctrl>-клавишу со стрелкой вправо	Ближайшую точку справа в направлении X на текущем графике.
<Ctrl>-клавишу со стрелкой вверх	Ближайшую точку сверху в направлении Y на текущем графике.
<Ctrl>-клавишу со стрелкой вниз	Ближайшую точку снизу в направлении Y на текущем графике.
<Home>	Первую видимую точку текущего графика.
<End>	Последнюю видимую точку текущего графика.
<Shift-Page UP>	Предыдущий график.
<Shift-Page Down>	Следующий график.

Если график сконфигурирован как элемент управления в горячем режиме, событие `commit` начнет генерироваться, если пользователь нажмет одну из вышеупомянутых клавиш.

Работа с графиком с помощью курсоров с использованием мышки:

- Для перемещения курсора нажмите его мышкой и потяните. Если курсор находится в режиме захвата точки, то он будет перемещаться за мышкой до тех пор, пока пользователь не отпустит кнопку мышки, а затем перепрыгнет на ближайшую точку данных. Если курсор находится в режиме произвольного формата, то он будет перемещаться за мышкой до тех пор, пока пользователь не отпустит кнопку мышки, и затем остановится в этом положении.
- Перемещайте активный курсор влево или вправо, перемещая мышкой маркер активного курсора у верхнего или нижнего края области графика. Таким же образом можно перемещать активный курсор вверх и вниз, если передвигать маркер активного курсора у левого или правого края области графика.

Если графический элемент управления сконфигурирован в горячем режиме, событие `commit` начнет генерироваться, если пользователь передвинет курсор с помощью клавиш со стрелками или отпустит кнопку мышки после перемещения курсора.

События `EVENT_VAL_CHANGED` непрерывно генерируются, пока пользователь перемещает графический курсор.

Элементы управления ленточными схемами

Элементы управления ленточными схемами выдают графические данные в реальном времени. Ленточная схема состоит из одной или более кривых, которые обновляются одновременно. На приведенном ниже рисунке представлен пример управления ленточной схемой.

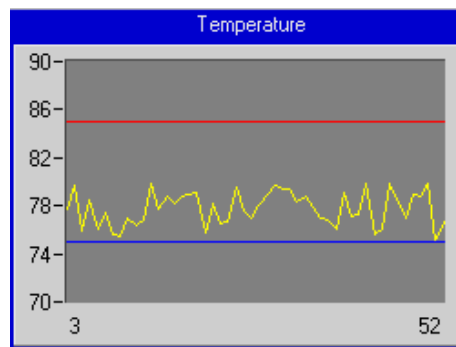


Рисунок 1-19. Управление ленточной схемой.

Элементы управления рисунками

Управление рисунками позволяет выводить на панель различные образы, такие как логотипы и диаграммы. Например, Вы можете использовать управление рисунками для введения схемы, инструктирующей пользователя, как выбрать (подключить) блок для тестирования. Если Вы хотите установить образы в командные кнопки или кнопки переключения, или в элементы управления посылкой вызова, LabWindows/CVI освобождает Вас от необходимости программирования простого элемента управления Рисунком. Для рисунков имеются отдельные командные кнопки, кнопки переключения и элементы управления кольцевым списком.

В Таблице 1-2 приведены форматы образов, работающие со всеми типами элементов управления рисунками.

Таблица 1-2. Форматы образов.

Формат образа	Платформа
PCX	Windows и UNIX
BMP, DIB, RLE, ICO	только Windows
XWD	только UNIX

Таймеры

Используйте *таймеры* для запуска определенных действий через заданные интервалы времени. Таймер освобождает Вас от необходимости планирования этих действий. На Рисунке 1-20 приведен таймер.



Рисунок 1-20. Таймер

Таймеры повторяют заданное действие через заданные временные интервалы в течение неопределенного периода времени, что делает их полезными инструментами программирования в случаях, когда требуется использование повторяющихся действий. Вы можете задать вызов функции в конце каждого интервала.

LabWindows/CVI имеет функции задержки и перезадавания таймеров.

Таймеры невидимы во время выполнения программы. Они видимы только в Редакторе пользовательского интерфейса.

Использование всплывающих панелей

Всплывающая панель представляет собой панель, которая появляется, принимает пользовательский ввод, а затем исчезает.

Всплывающие панели могут быть магазинного типа, когда каждая новая всплывающая панель появляется поверх предыдущей. Когда всплывающая панель активна, она появляется на переднем плане и является на данный момент единственной панелью, с которой Вы можете работать.

Библиотека пользовательского интерфейса содержит набор заданных всплывающих панелей для общих операций, таких как выдача многострочных сообщений, запрос пользовательского ввода, запрос подтверждения пользователем, выбор файла и графическая презентация цифровых данных.

Каждая заданная всплывающая панель вызывается определенными функциональными вызовами. Функция выводит на экран всплывающую панель и ожидает выбора пользователем определенной операции. Затем панель исчезает и управление возвращается к программе.

Работа с всплывающей панелью сообщений

Всплывающая панель сообщений выдает многострочные сообщения. Для начала новой строки текста используйте символ новой строки ($\backslash n$). На приведенном ниже рисунке представлен пример всплывающей панели сообщений.



Рисунок 1-21. Всплывающая панель сообщений.

Работа с всплывающей панелью обобщенных сообщений

Всплывающая панель обобщенных сообщений выводит на экран всплывающую панель со строкой сообщения, буфером отклика и не более, чем с тремя кнопками с программируемыми метками.

Работа с всплывающей панелью приглашений

Всплывающая панель приглашений запрашивает ввод пользователя. Пример такой панели приведен на Рисунке 1-22.

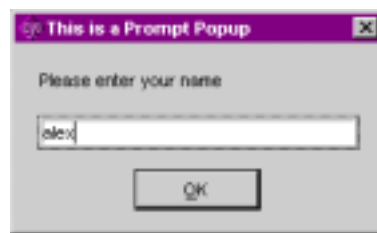


Рисунок 1-22. Всплывающая панель приглашений.

Работа с всплывающей панелью подтверждения

Всплывающая панель подтверждения позволяет пользователю подтвердить действие. Пример такой панели приведен на Рисунке 1-23.

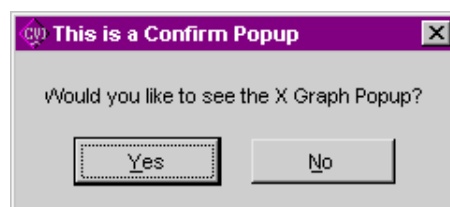


Рисунок 1-23. Всплывающая панель подтверждения.

Работа с всплывающей панелью выбора файла

Всплывающая панель выбора файла выдает список файлов, имеющихся на диске, из которых пользователь может выбрать требуемый файл. Всплывающая панель выбора файла имеет следующий вид.

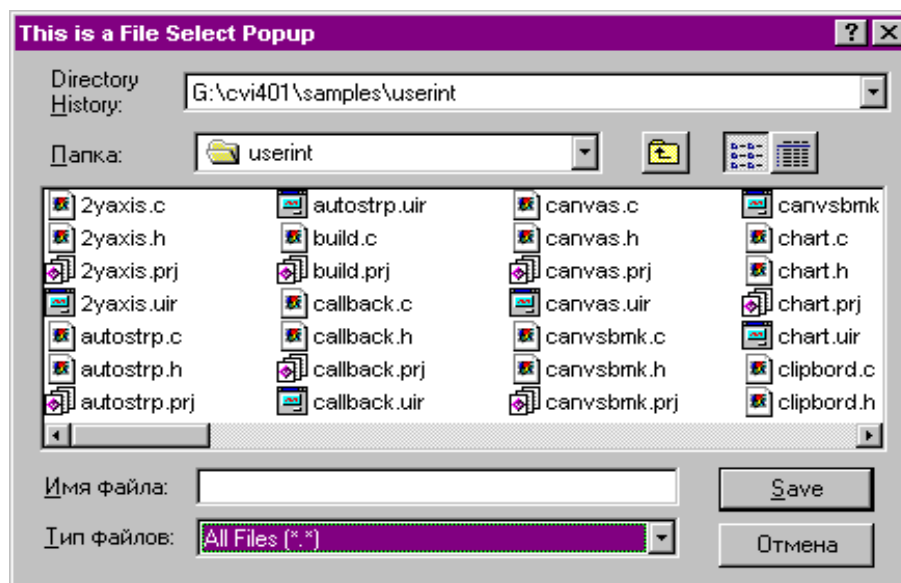


Рисунок 1-24. Всплывающая панель выбора файла.

Уникальной особенностью данного диалогового окна является кольцевой список **Directories** в верхнем правом углу диалогового окна. При активизации, данный кольцевой список дает пользователю возможность сделать выбор из списка директорий, содержащих ранее открытые файлы.

Работа с графической всплывающей панелью

Для выдачи графика цифровых данных Вы можете выбрать один из четырех различных типов графических всплывающих панелей: *X-graph pop-up*, *Y-graph pop-up*, *XY-graph pop-up* и *Waveform graph pop-up*. На Рисунке 1-25 приведен пример графической всплывающей панели.

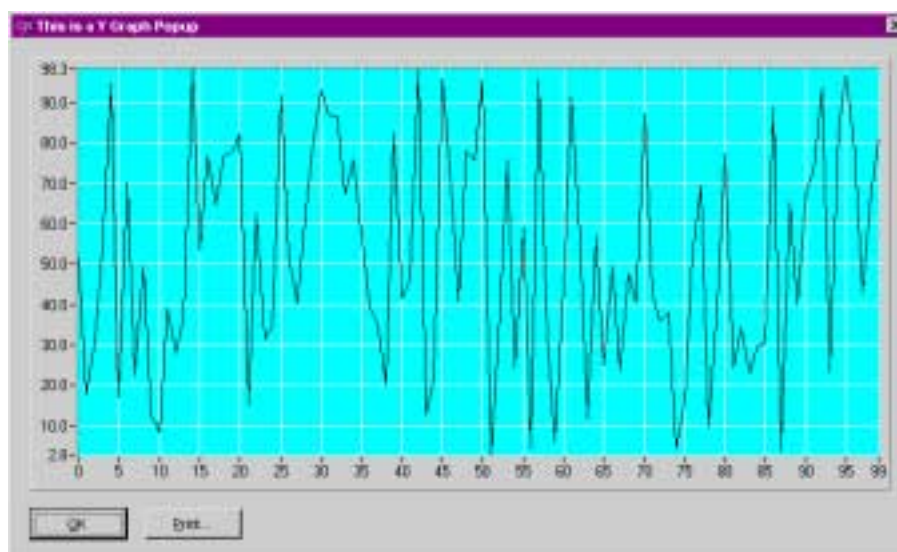


Рисунок 1-25. Графическая всплывающая панель.

Использование шрифтов

Меташрифты, использующие шрифты, типичные для каждой платформы

Меташрифты содержат информацию о шрифтах, размер точки и текстовые стили, например, полужирный, подчеркивание, курсив и перечеркивание. Приведенные ниже меташрифты, использующие шрифты, типичные для каждой платформы, поддерживаются LabWindows/CVI. Каждый из этих меташрифтов задавался так, чтобы его высота и ширина максимально совпадали для всех платформ.

```
VAL_MENU_META_FONT  
VAL_DIALOG_META_FONT  
VAL_EDITOR_META_FONT  
VAL_APP_META_FONT
```

Вы можете создавать меташрифты на базе любого шрифта Вашей системы с помощью функции CreateMetaFont.

Шрифты, использующие типы шрифтов, типичные для каждой платформы

Приведенные ниже шрифты используют типы шрифтов, типичные как для системы PC, так и для системы UNIX. Шрифты включают в себя только информацию о типах шрифтов.

```
VAL_MENU_FONT  
VAL_DIALOG_FONT  
VAL_EDITOR_FONT  
VAL_APP_FONT
```

Меташрифты, использующие типы шрифтов, инсталлированные LabWindows/CVI

Приведенные ниже меташрифты включены в LabWindows/CVI, но в них использованы типы шрифтов, не типичные для систем PC или UNIX. Типы шрифтов поставляются вместе с LabWindows/CVI и исполняющей машиной.

```
VAL_7SEG_META_FONT  
VAL_SYSTEM_META_FONT
```

Глава 2. Справочник по Редактору пользовательского интерфейса

Как было описано в Главе 1, Вы можете создавать свой GUI программно, используя вызовы функций, или интерактивно, используя Редактор пользовательского интерфейса. В настоящей главе описан вариант интерактивного создания GUI. В ней представлено описание Редактора пользовательского интерфейса и процедуры по созданию и редактированию панелей, элементов управления и линеек меню.

При использовании Редактора пользовательского интерфейса Вы создаете и модифицируете ресурсные файлы пользовательского интерфейса (.uir). Войдите в Редактор пользовательского интерфейса, выбрав опцию **New** или **Open** из меню **File** и выбрав **User Interface (*.uir)**.

Общая информация о Редакторе пользовательского интерфейса

Окно Редактора пользовательского интерфейса имеет следующий вид.

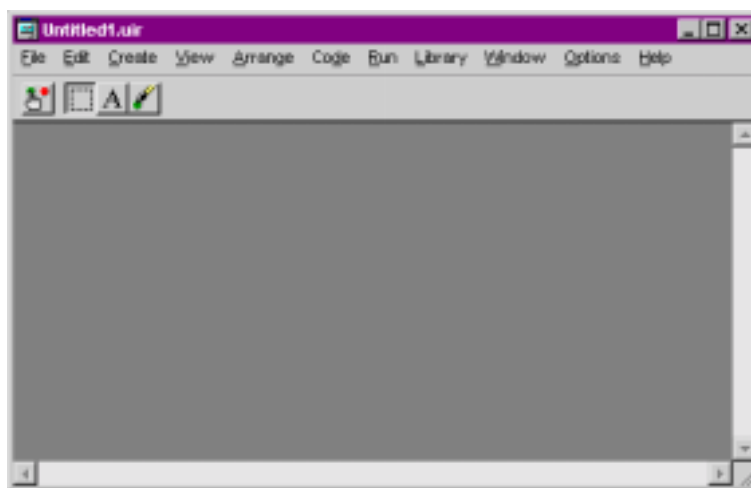


Рисунок 2-1. Окно Редактора пользовательского интерфейса.

Из данного окна Вы можете создавать и редактировать панели, элементы управления и линейки меню GUI. Описание меню дано в разделе *Меню Редактора пользовательского интерфейса* в данной главе. Используйте линейку инструментов, расположенную под линейкой меню, для высокоуровневого редактирования с помощью мышки. Когда Вы нажмете кнопку мышки, наведя ее на любой инструмент, курсор мышки изменится, отражая новый режим редактирования.



Используя инструмент Редактирования, Вы можете выбрать, установить и измерить размер объектов.



Используя инструмент Маркирования, Вы можете модифицировать текст, связанный с объектами.



Используя инструмент Задания цвета, Вы можете задать для объектов определенный цвет. Нажатие правой кнопки мышки выведет на экран цветовую палитру, из которой Вы сможете выбрать желаемый цвет. Нажатие левой кнопки мышки автоматически окрасит объект в текущий цвет инструмента Задания цвета. Если Вы нажмете клавишу <Ctrl>, не отпуская ее, то инструмент изменится на иконку пипетки. Если Вы нажмете на объект иконкой пипетки, то текущий цвет инструмента Задания цвета изменится на цвет данного объекта. Затем Вы сможете применить цвет этого объекта к другому объекту.



Используйте Операционный инструмент для проведения операций с объектами. Если Вы находитесь в операционном режиме, события выводятся на экран в правой части линейки инструментов. Эти дисплеи событий имеют встроенную задержку, что дает Вам возможность увидеть каждое событие.

Использование всплывающих меню Редактора пользовательского интерфейса

Всплывающее меню можно вызвать, нажав правой кнопкой мышки на Окно Редактора пользовательского интерфейса. Тип возникшего меню зависит от места, на которое Вы нажали.

- Если Вы нажмете мышкой на задний план Окна Редактора пользовательского интерфейса, то появится всплывающее меню с командами для создания панели или линейки меню.
- Если Вы нажмете мышкой на задний план панели, то появится меню со всеми типами элементов управления, которые Вы можете создать.
- Если Вы нажмете мышкой на элемент управления, то появится меню с командами генерирования или просмотра функции обратного вызова для данного элемента управления.

Общая информация о Построителе кодов

С помощью Построителя кодов LabWindows/CVI Вы можете автоматически создавать полный код C, который компилируется и работает на основе файла пользовательского интерфейса (.uir), который Вы создаете или редактируете. Выбрав определенные опции из предложенных в меню **Code**, Вы можете создать *“скелет” программы*. “Скелет” программы представляет собой синтаксически и программно корректный код, который компилируется и выполняется до того, как Вы введете хотя бы одну строку кода. Благодаря возможностям Построителя кодов, Вы сберегаете время, не вводя стандартный код, включенный в каждую программу, исключаете синтаксические ошибки и опечатки и поддерживаете организацию файла исходного кода в соответствии со стилем программирования. Так как программа Построителя кодов компилируется и выполняется одновременно, Вы можете разрабатывать и испытывать создаваемый Вами проект, концентрируя свое внимание на каждой функции по очереди.

Если Вы выбираете **Code » Generate » All Code**, LabWindows/CVI установит операторы #include, декларации переменных, планы функций обратного вызова и функцию main в файл исходного кода, заданный Вами как целевой файл. Каждый скелет функции включает в себя конструкцию переключателя с оператором выбора для каждого задаваемого Вами события по умолчанию. Выбрав **Code » Preferences**, Вы можете установить события по умолчанию для функций обратного вызова элементов управления и функций обратного вызова панелей. Хотя скелет программы и выполняется, Вы должны привести его в соответствие с тем, какие действия, по Вашему мнению, должны выполняться в случае возникновения каждого из событий.

Если Вы генерируете код для функции обратного вызова определенного элемента управления или панели, то LabWindows/CVI вводит скелет программы для данной функции в целевой файл в том же полном формате, как это было сделано для команды **Code » Generate » All Code**. Однако, данный код может не выполняться. Для того, чтобы проект выполнялся, необходима функция main. Если у Вас будет отсутствовать функция main или любая из функций обратного вызова, определенных в файле .uir, код окажется незавершенным.

Хорошим решением, по нашему мнению, будет использование сначала опции **Code » Generate » All Code**, чтобы создать выполняемый проект из текущего состояния файла .uir. Затем, после добавления в файл .uir панелей, элементов управления или элементов меню, используйте команды **Generate Panel, Control Callbacks** и **Menu Callbacks**, чтобы сделать соответствующие добавления в исходный файл.

Кроме того, с помощью Построителя кодов Вы можете организовать завершение программы для автоматически сгенерированного кода. Для того, чтобы программа Построителя кодов завершилась успешно, Вы должны включить вызов в функцию `QuitUserInterface`. Когда Вы выбираете **Code » Generate » All Code**, диалоговое окно `Generate All Code` попросит Вас выбрать, какие функции обратного вызова будут завершать программу. Чтобы обеспечить корректное завершение программы, Вы можете выбрать одну или более функций обратного вызова.

Меню Редактора пользовательского интерфейса

Редактор пользовательского интерфейса имеет линейку меню, содержащую следующие опции: **File, Edit, Create, View, Options, Window** и **Help**.

Меню File

На приведенном ниже рисунке представлено меню **File**.

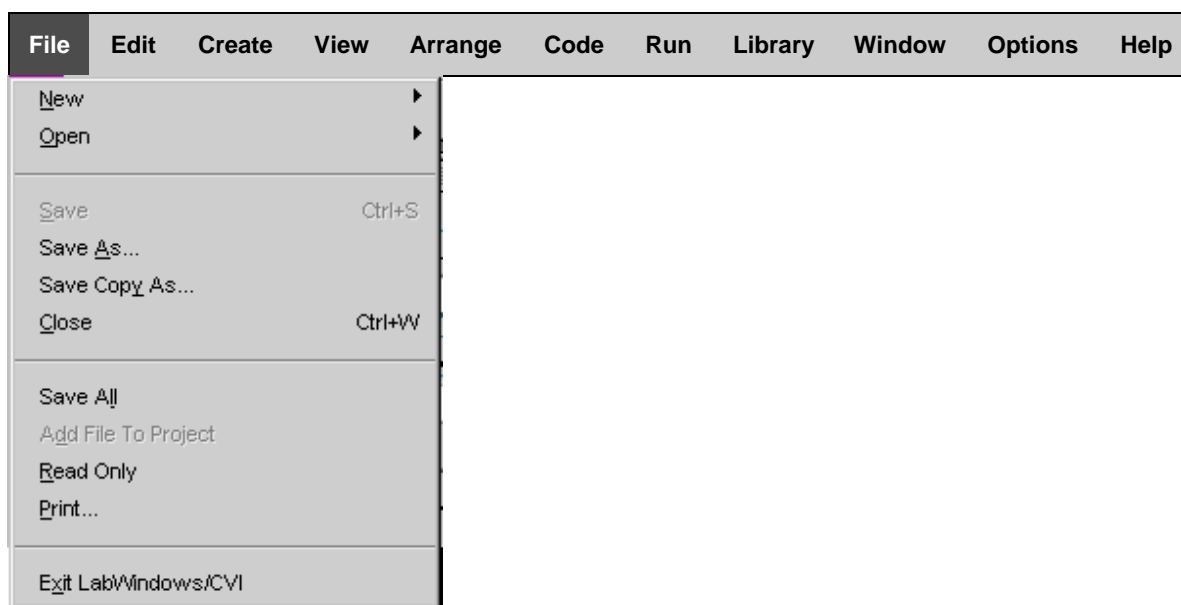


Рисунок 2-2. Меню File.

Команды **New, Open, Save** и **Exit LabWindows/CVI**

Команды **New, Open, Save** и **Exit LabWindows/CVI** в линейке меню Редактора пользовательского интерфейса действуют так же, как и аналогичные команды в окне `Project`.

Команды **Save As** и **Close**

Команды **Save As** и **Close** действуют так же, как и команды **Save As** и **Close** в стандартных приложениях для Windows. Предполагается, что Вы уже знакомы с этими командами.

Save Copy As

Команда **Save Copy As** записывает содержание активного окна на диск, используя заданное пользователем имя, без изменения имени активного окна. Если Вы хотите присоединить другое расширение, введите новое расширение после имени файла. Если Вы не хотите присоединять расширение, то после имени файла введите только точку.

Save All

Команда **Save All** сохраняет все открытые файлы на диск.

Add File to Project

Команда **Add File to Project** добавляет файл `.uir` в текущее окно списка проекта.

Read Only

Команда **Read Only** блокирует возможности редактирования в текущем окне.

Print

Команда **Print** открывает диалоговое окно **Print**, позволяющее Вам посылать весь файл `.uir` или видимую область экрана на принтер или в файл. Диалоговое окно позволяет также установить привилегии печати.

Меню Edit

Элементы меню **Edit** используются для редактирования панелей, элементов управления и линеек меню. На Рисунке 2-3 приведен пример меню **Edit**.

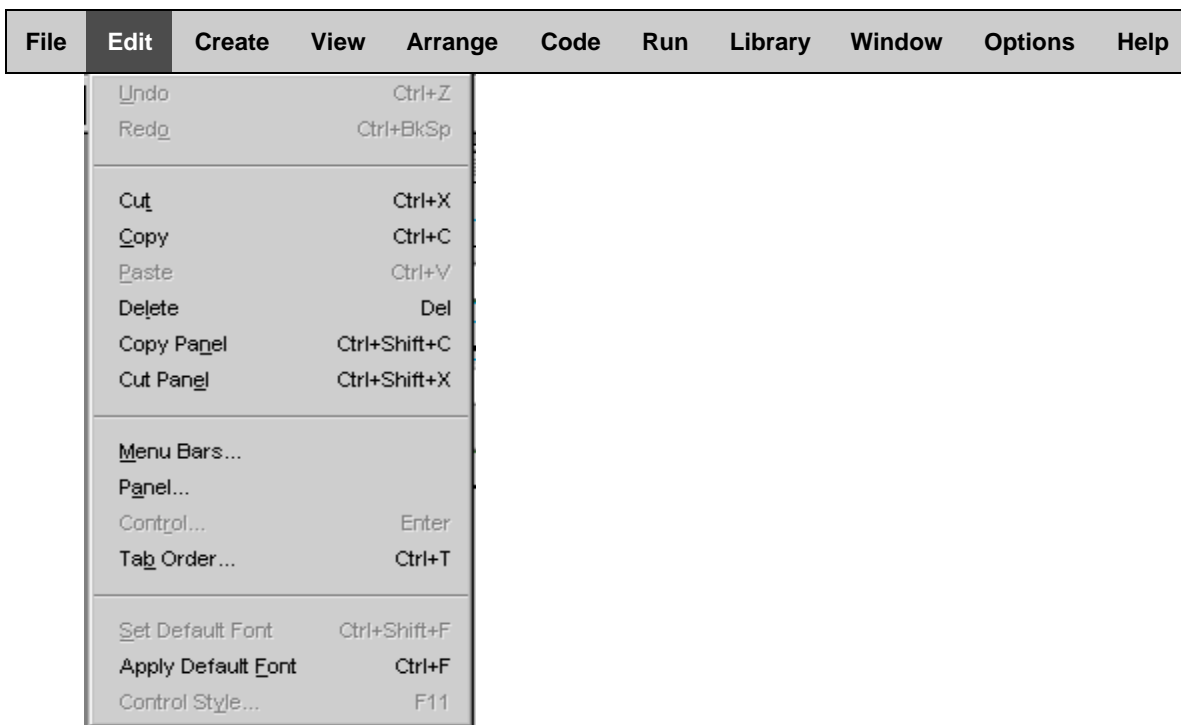


Рисунок 2-3. Меню Edit.

Внимание: *Undo и Redo разблокировываются, когда Вы выполняете редактирование. Команды Cut и Copy разблокировываются, когда Вы выбираете элемент управления, а Paste - когда Вы вводите объект в Буфер вырезанного изображения, используя команду Cut или Copy. Если Вы выбираете заблокированную в данный момент команду редактирования, то ничего не произойдет.*

Undo и Redo

Команда **Undo** отменяет Ваше последнее действие редактирования и экран возвращается в предыдущее состояние. Действия редактирования сохраняются в стеке, так что Вы можете отменить любую серию своих действий по редактированию. Стек сохраняет до 100 действий редактирования и устанавливается с помощью команды **Preferences** в меню **Options**.

Команда **Redo** отменяет Вашу последнюю команду **Undo**, восстанавливая экран в его предыдущем состоянии. **Redo** полезна, когда Вы используете команду **Undo** для отмены серии действий по редактированию и случайно отменяете больше действий, чем было необходимо. Команда **Redo** доступна только тогда, когда Вашей предыдущей операцией было **Undo**. Любое действие блокирует команду **Redo**.

Действия, которые могут быть аннулированы и восстановлены, динамически выдаются в меню, например, когда Вы передвигаете элемент управления, меню представляет опцию **Undo Move Control**.

Cut и Copy

Команды **Cut** и **Copy** вводят элементы управления в Буфер вырезанного изображения. Команда **Cut** удаляет выбранный элемент управления и устанавливает его в Буфер вырезанного изображения. Команда **Copy** копирует выбранный элемент управления и устанавливает его в Буфер вырезанного изображения, оставляя выбранный элемент управления в его оригинальном месте. Вырезаемые или копируемые Вами элементы управления не накапливаются в Буфере вырезанного изображения. Каждый раз, когда Вы вырезаете или копируете элемент управления, он заменяет собой предыдущий элемент управления, сохраненный в Буфере вырезанного изображения.

Последовательность шагов при использовании команд **Cut** или **Copy**:

1. Выбрать элемент управления, который Вы хотите ввести в Буфер вырезанного изображения, нажав на него клавишей мышки или нажимая <Tab> до тех пор, пока данный элемент управления не высветится. Несколько элементов управления можно выбрать, если нажать на первый из них мышкой и, не отпуская кнопку, передвинуть мышку так, чтобы высветились все необходимые элементы управления, или же если нажать на требуемые элементы управления параллельно с нажатием <Shift>.
2. Выбрать **Cut** и **Copy** из меню **Edit**.

Paste

Команда **Paste** вставляет элементы управления, панели или текст из Буфера вырезанного изображения. С помощью **Paste** Вы можете вставить объект из Буфера вырезанного изображения любое количество раз. Элементы управления и панели остаются в Буфере до тех пор, пока Вы снова не используете команды **Cut**, **Cut Panel**, **Copy** или **Copy Panel**. Команды **New** и **Open** не стирают информацию в Буфере вырезанного изображения.

Delete

Команда **Delete** удаляет выбранные элементы управления, не вводя их в Буфер вырезанного изображения. Так как **Delete** не вводит элементы управления в Буфер вырезанного изображения, Вы не сможете восстановить их с помощью команды **Paste**.

Copy Panel и Cut Panel

Команды **Copy Panel** и **Cut Panel** вводят всю панель в Буфер вырезанного изображения. Команда **Cut Panel** удаляет выбранную панель и вводит ее в Буфер вырезанного изображения. Команда **Copy Panel** копирует выбранную панель и вводит ее в Буфер вырезанного изображения, оставляя выбранную панель в ее оригинальном месте. Вырезаемые или копируемые Вами панели не накапливаются в Буфере вырезанного изображения. Каждый раз, когда Вы вырезаете или копируете панель, она заменяет собой предыдущую панель, сохраненную в Буфере вырезанного изображения.

Последовательность шагов при использовании команд **Copy Panel** или **Cut Panel**.

1. Выбрать панель, которую Вы хотите ввести в Буфер вырезанного изображения, нажав на нее мышкой или нажимая <Shift-Ctrl> и одну из клавиш со стрелками (вверх, вниз, вправо, влево) до тех пор, пока панель не станет активной.
2. Выбрать **Copy Panel** или **Cut Panel** из меню **Edit**.

Menu Bars

Команда **Menu Bars** вызывает диалоговое окно со Списком линеек меню, показанное на рисунке ниже.

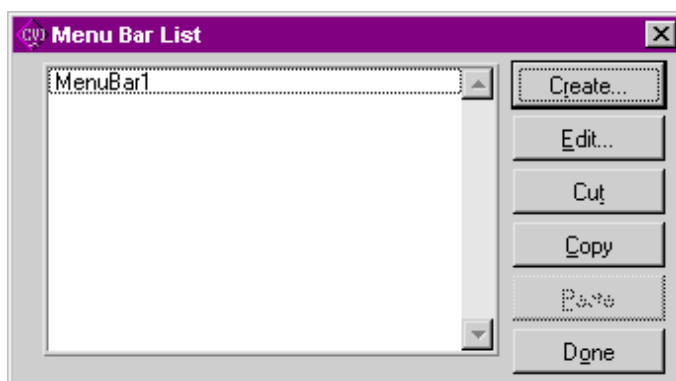


Рисунок 2-4. Диалоговое окно со списком линеек меню.

Список включает в себя все элементы меню, имеющиеся в ресурсном файле, записанные с постоянным префиксом. Ниже приведено описание командных кнопок.

- **Create** открывает новое диалоговое окно линейки меню Edit (рисунок 2-5). После создания, линейка меню устанавливается под текущей линейкой меню в списке линеек меню.
- **Edit** открывает диалоговое окно линейки меню Edit для выбранной линейки меню.
- **Cut** удаляет высвеченный в данный момент элемент в списке линеек меню и копирует его в буфер вырезанного изображения линеек меню.
- **Copy** копирует высвеченный в данный момент элемент в списке линеек меню в буфер вырезанного изображения линеек меню.
- **Paste** вводит содержание буфера вырезанного изображения линеек меню в список линеек меню. Линейка меню устанавливается над высвеченным в данный момент элементом в списке линеек меню.
- **Done** закрывает диалоговое окно Списка линеек меню.

На приведенном ниже рисунке представлено диалоговое окно линейки меню Edit.

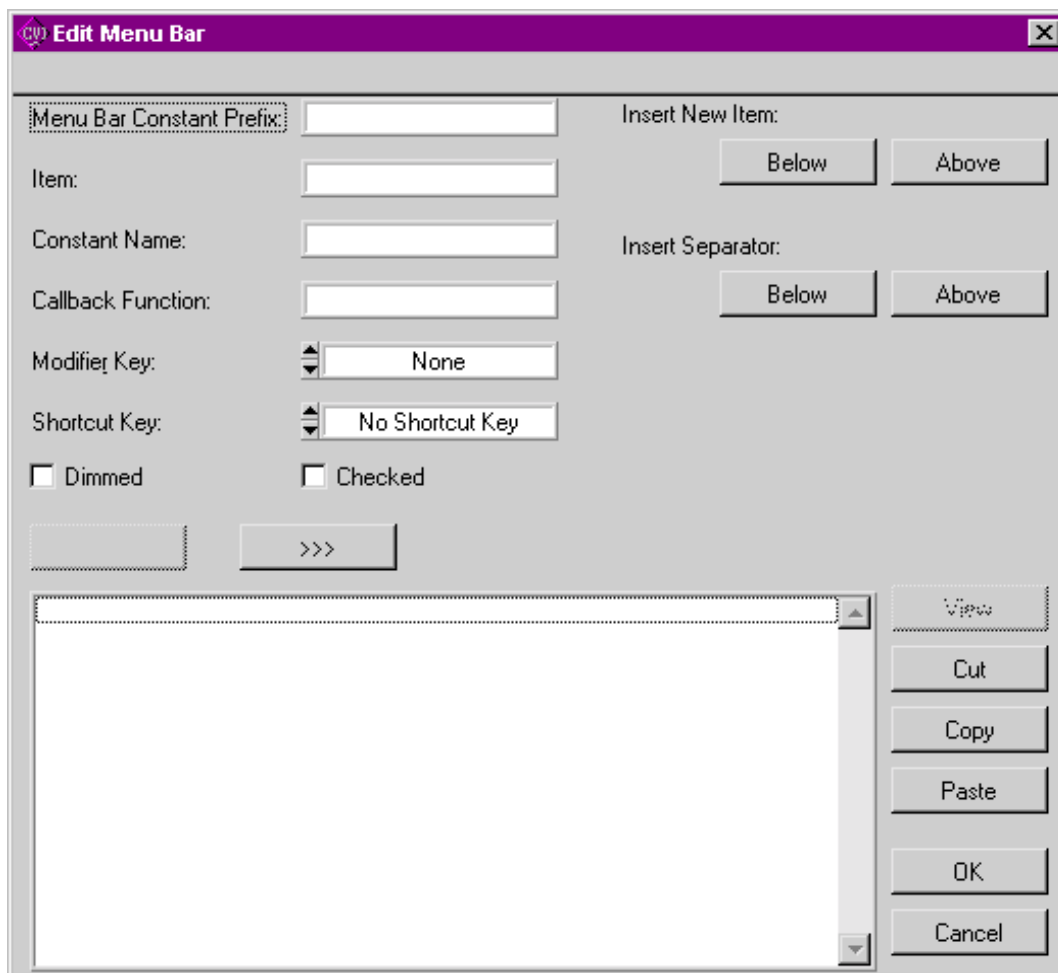

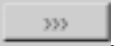


Рисунок 2-5. Диалоговое окно линейки меню Edit.

Диалоговое окно линейки меню Edit предоставляет следующие опции.

- **Menu Bar Constant Prefix** представляет собой идентификатор ресурса для линейки меню. Идентификатор ресурса передается функции `LoadMenuBar`, которая загружает панель в память. **Menu Bar Constant Prefix** определяется в файле `.h`, когда Вы сохраняете файл `.uir`. Если Вы не зададите **Menu Bar Constant Prefix**, Редактор пользовательского интерфейса задает ее сам, когда Вы сохраняете файл `.uir`.
- **Item** показывает имя текущего меню, подменю или команды меню. Если перед любой буквой элемента Вы введете две подчеркивающие линии, то пользователь сможет выбрать элемент, нажав `<Alt>` и подчеркнутую букву.
- **Constant Name** присоединяется к **Menu Bar Constant Prefix**, образуя идентификатор текущего элемента. Идентификатор используется в таких функциях, как `GetMenuBarAttribute` и `SetMenuBarAttribute`. Кроме того, `GetUserEvent` возвращает данный идентификатор, если событие генерируется командой меню.
- Название **Callback Function** является необязательным. В данное окно Вы можете ввести имя функции, которая будет вызываться при генерировании события текущим элементом меню.
- Сочетание **Modifier Key** и **Shortcut Key** образует горячую клавишу для текущего элемента.
- **Dimmed** задает, будет ли текущий элемент изначально не высвеченным.

- **Checked** задает, будет ли у текущего элемента изначально метка проверки.
- **Insert New Item** вставляет следующий элемент над или под выбранным текущим элементом.
- **Insert Separator** вставляет разделитель (черту) над или под выбранным текущим элементом.
- Левая иерархическая кнопка, , перемещает выбранный текущий элемент на один уровень вверх в иерархии подменю.
- Правая иерархическая кнопка, , перемещает выбранный текущий элемент на один уровень вниз в иерархии подменю.
- **View** выдает текущее состояние линейки меню и выпадающих меню.

Panel

Команда **Panel** открывает диалоговое окно Panel Edit. Данное диалоговое окно имеет три раздела, озаглавленные *Source Code Connection* (Связь с исходным кодом), *Panel Attributes* (Атрибуты панели) и *Quick Edit Window* (Окно быстрого редактирования).

Ниже представлен раздел Связь с исходным кодом диалогового окна панели Edit.

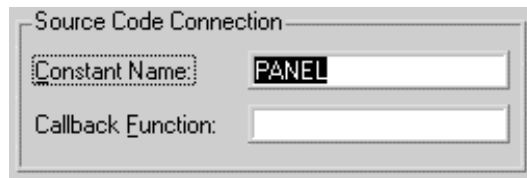


Рисунок 2-6. Связь с исходным кодом.

В окне **Constant Name** Вы вводите идентификатор ресурса для панели. Вы передаете этот идентификатор ресурса функции `LoadPanel` для загрузки панели в память. **Constant Name** определяется в файле `.h` когда Вы сохраняете файл `.uir`. Если Вы не зададите **Constant Name**, то Редактор пользовательского интерфейса сам задаст его во время сохранения Вами файла `.uir`.

Задание **Callback Function** является необязательным. В данном окне Вы можете ввести имя функции, которая будет вызываться при генерировании на панели события.

Раздел Panel Settings (Установки панели) диалогового окна панели Edit имеют следующий вид.

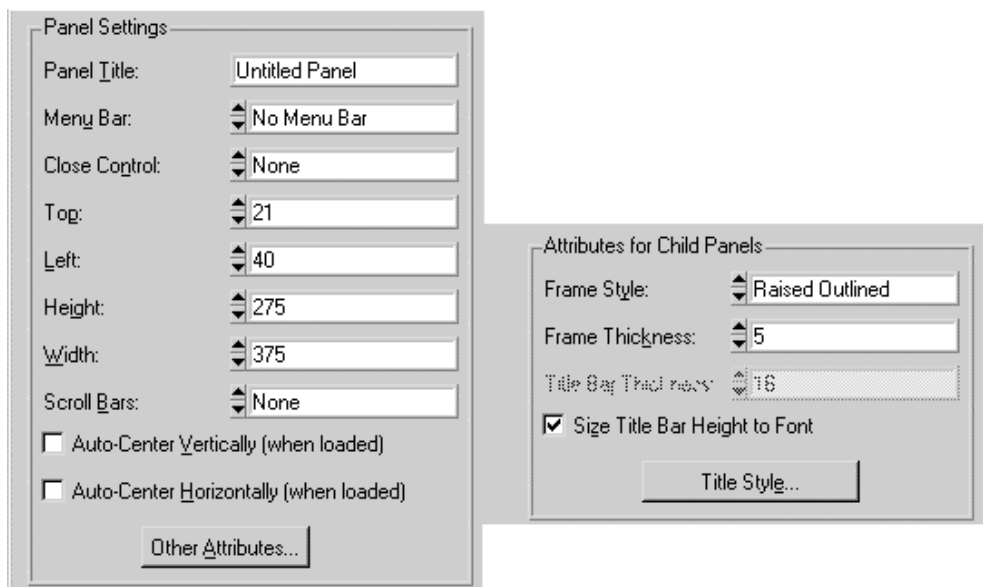


Рисунок 2-7. Атрибуты панелей.

Внимание: На приведенном выше рисунке представлены установки и атрибуты, имеющиеся в версии Windows LabWindows/СVI. Диалоговое окно панели Edit в версии UNIX LabWindows/СVI не предоставляет некоторые элементы, т.к. данная программа не может обновлять некоторые атрибуты панели для родительских панелей.

Раздел Окно быстрого редактирования диалогового окна панели Edit показан на приведенном ниже рисунке.

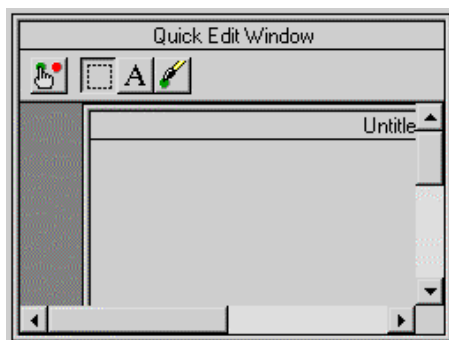


Рисунок 2-8. Окно быстрого редактирования.

Из окна Быстрого редактирования Вы можете выполнять на панели операции по высокоуровневому редактированию. Инструменты, представленные на линейке инструментов, действуют так же, как и в основном окне Редактора пользовательского интерфейса. За дополнительной информацией обращайтесь к разделу *Общая информация о Редакторе пользовательского интерфейса* в данной главе.

Control

Команда **Control** открывает диалоговое окно для редактирования выбранного элемента управления. Это диалоговое окно Вы также можете открыть двойным нажатием мышкой на требуемый элемент управления. Диалоговое окно обычно имеет пять разделов, озаглавленных *Source Code Connection* (Связь с исходным кодом), *Control Settings* (Установки элементов управления), *Control Appearance* (Вид элемента управления), *Quick Edit Window* (Окно быстрого редактирования) и *Label Appearance* (Вид метки). Содержание разделов слегка варьируется в зависимости от редактируемого элемента управления.

Раздел Связь с исходным кодом диалогового окна элементов управления представлен на Рисунке 2-9.

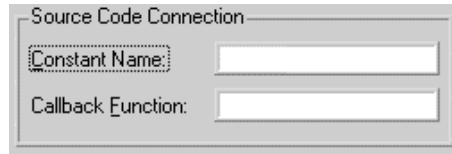


Рисунок 2-9. Связь с исходным кодом.

Для создания идентификатора элемента управления Редактор пользовательского интерфейса присоединяет **Constant Name** к идентификатору ресурса панели. Полученный идентификатор идентифицирует элемент управления в любой специфичной для данного элемента управления функции, такой как `GetCtrlVal` и `SetCtrlAttribute`. Идентификатор будет определен в файле `.h` во время сохранения Вами файла `.uir`. Если Вы не зададите **Constant Name**, то Редактор пользовательского интерфейса задаст его сам при сохранении Вами файла `.uir`.

Задание **Callback Function** является необязательным. В данном окне Вы можете ввести имя функции, которая будет вызываться при генерировании на панели сообщения о событии.

Раздел диалогового окна Установки элемента управления воспринимает тип редактируемого элемента управления. Он содержит атрибуты данных для этого элемента управления. На приведенном ниже рисунке представлен раздел Установки элемента управления для Цифрового элемента управления.

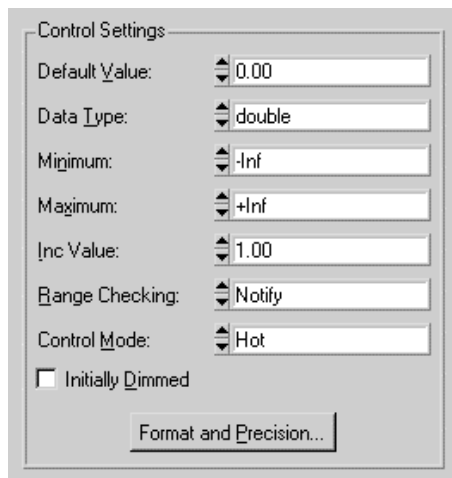


Рисунок 2-10. Установки элемента управления для Цифрового элемента управления.

Окна списков и кольцевых списков имеют в разделе Установки элемента управления кнопку **Label/Value Pairs**. Данная кнопка активизирует диалоговое окно **Edit Label/Value Pairs**, представленное на Рисунке 2-11.

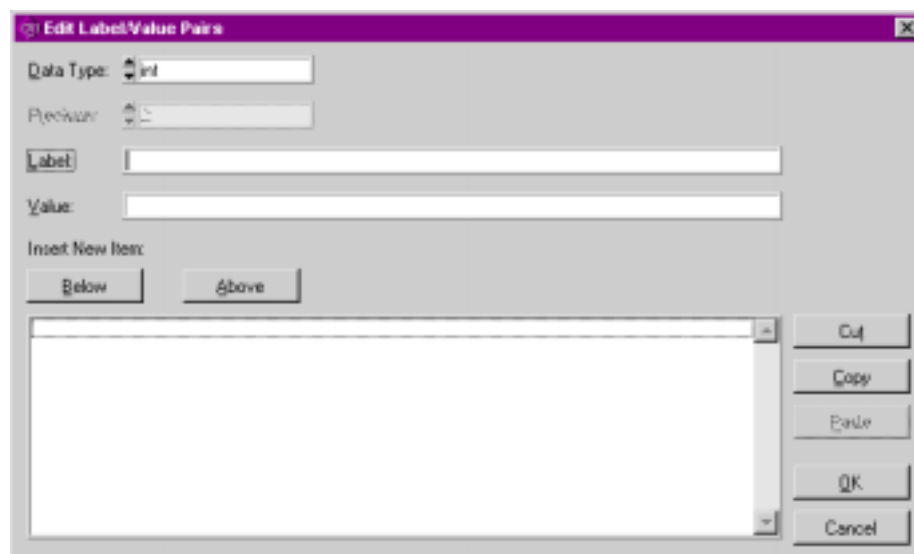


Рисунок 2-11. Диалоговое окно Edit Label/Value Pairs.

Используйте диалоговое окно Edit Label/Value Pairs для создания и редактирования содержания элементов управления в окнах списков и кольцевых списках. Используйте функции элементов управления списком в Библиотеке пользовательского интерфейса для управления окнами списков и кольцевых списков.

Раздел диалогового окна Вид элементов управления воспринимает тип редактируемого элемента управления. Он содержит атрибуты, относящиеся к физическому виду элемента управления. На Рисунке 2-12 приведен раздел Вид элемента управления для Цифрового элемента управления.

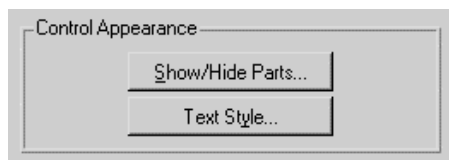


Рисунок 2-12. Вид элемента управления для Цифрового элемента управления.

Раздел Вид метки диалогового окна содержит атрибуты, относящиеся к физическому виду метки элемента управления. На Рисунке 2-13 показан раздел Вид метки для Цифрового элемента управления.

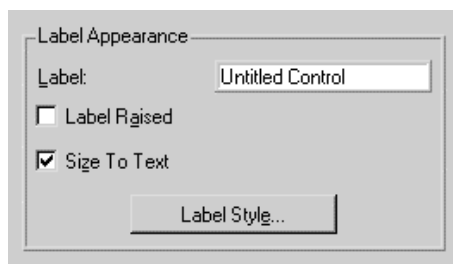


Рисунок 2-13. Вид метки для Цифрового элемента управления.

Если Вы введете две подчеркивающие черты перед любой буквой в метке, то пользователь сможет выбрать данный элемент управления, нажав одновременно <Alt> и подчеркнутую букву, при условии, когда линейка меню недоступна. Данное свойство позволяет Вам получить доступ к элементам управления на всплывающих панелях.

Раздел Окно быстрого редактирования диалогового окна элемента управления Edit показан на приведенном ниже рисунке.

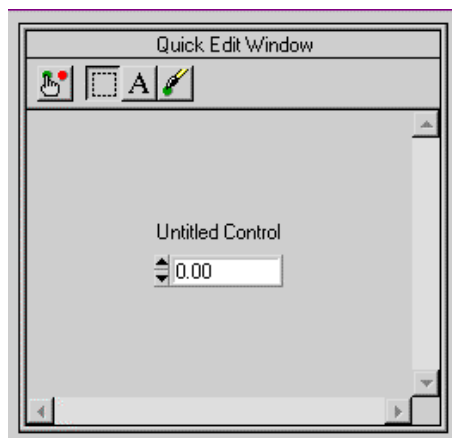


Рисунок 2-14. Окно быстрого редактирования.

Из окна Быстрого редактирования Вы можете выполнять высокоуровневые операции по редактированию элемента управления. Инструменты, представленные в линейке инструментов, работают подобно инструментам в основном окне Редактора пользовательского интерфейса. Кроме того, Окно быстрого редактирования обеспечивает мгновенную обратную связь при любых изменениях, произведенных в другом разделе диалогового окна.

Проще говоря, диалоговое окно любого элемента управления позволяет Вам интерактивно устанавливать все атрибуты элемента управления.

Порядок табуляции

Каждый элемент управления на панели имеет свое положение в порядке табуляции. Порядок табуляции определяет, какой элемент управления будет активизироваться следующим при нажатии пользователем <Tab> или <Shift-Tab>.

Когда Вы создаете элемент управления, он добавляется в конец списка порядка табуляции. Если Вы копируете и вставляете элемент управления, он устанавливается непосредственно перед исходным элементом управления. Для ввода панели в режим редактирования порядка табуляции выберите **Tab Order** из меню **Edit**, как показано на Рисунке 2-15.

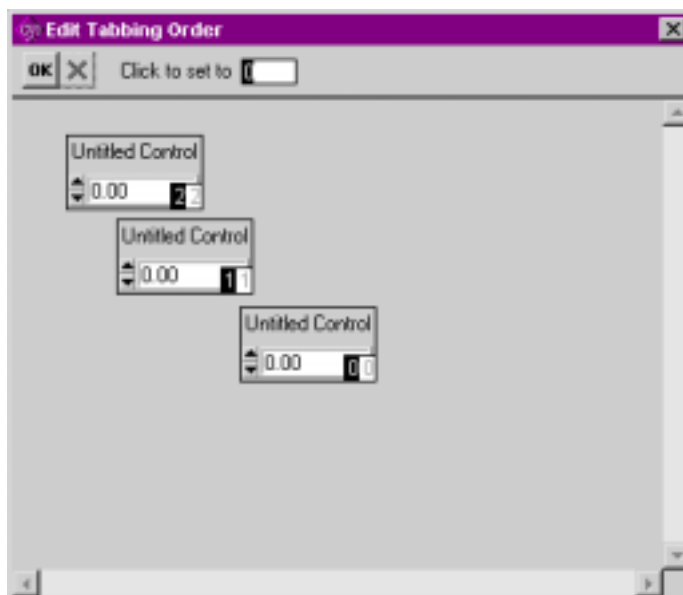


Рисунок 2-15. Диалоговое окно редактирования порядка табуляции.

Для изменения положения табуляции элемента управления на номер, указанный в окне **Click to set to**, нажмите на элемент управления специальным курсором с изображением руки. Вы можете изменить этот курсор на специальный курсор с изображением пипетки, нажав клавишу <Ctrl>. Курсор в виде пипетки изменяет номер в окне **Click to set to** на текущее положение табуляции, связанное с данным элементом управления. Нажав на **OK**, Вы установите новый порядок табуляции. Нажав на **Cancel**, Вы сотрете новый и восстановите оригинальный порядок табуляции, который появится в затененном виде справа от введенного Вами нового порядка табуляции.

Set Default Font

Команда **Set Default Font** в меню **Edit** делает шрифт выбранного текущего элемента управления шрифтом, принимаемым по умолчанию для всех элементов управления. Если метка также была выбрана (или была выбрана только метка), то шрифт данной метки становится принимаемым по умолчанию шрифтом для всех меток. Выбранные таким образом шрифты становятся принимаемыми по умолчанию и для вновь создаваемых элементов управления.

Apply Default Font

Команда **Apply Default Font** в меню **Edit** устанавливает шрифт выбранного текущего элемента управления (и/или метки) в качестве шрифта, принимаемым по умолчанию для всех элементов управления (и/или меток).

Control Style

Используйте команду **Control Style** для изменения стиля выбранного элемента управления. Например, Вы можете изменить элемент управления набором в виде ручки в ползунковый, при этом пары метка/значение останутся нетронутыми.

Меню Create

Команды в меню **Create** создают панели, линейки меню и элементы управления.

На Рисунке 2-16 показано меню **Create**.

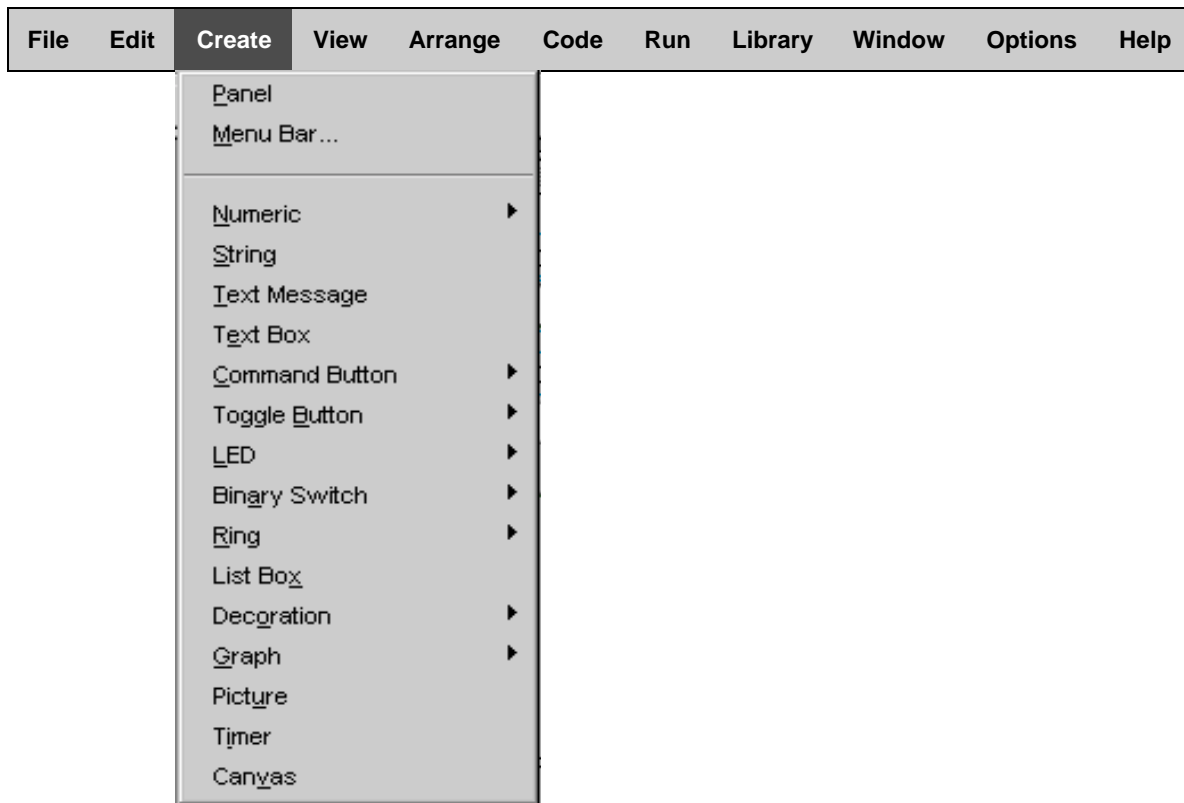


Рисунок 2-16. Меню Create.

Panel

Команда **Panel** устанавливает новую, незаглавленную панель в окно Редактора пользовательского интерфейса. В разделе *Меню Edit* данной главы представлена детальная информация по редактированию панели.

Menu Bar...

Команда **Menu Bar** открывает диалоговое окно Edit Menu Bar (Линейка меню). В разделе *Меню Edit* данной главы представлена детальная информация по редактированию меню.

Элементы управления

Остальные опции в меню **Create** позволяют Вам создавать элементы управления GUI. Вы можете отредактировать созданные Вами элементы управления из меню **Edit**. Детальная информация о редактировании меню представлена в разделе *Меню Edit* в данной главе.

Меню View

В настоящем разделе дано объяснение, как использовать команды меню **View** в окне Редактора пользовательского интерфейса.

На Рисунке 2-17 представлено меню **View**.

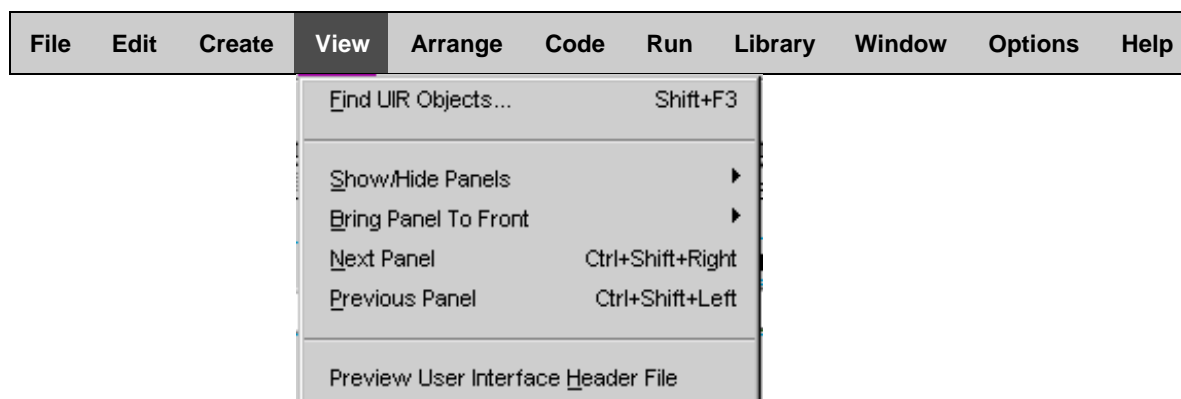


Рисунок 2-17. Меню View.

Find UIR Objects...

Используйте команду **Find UIR Objects** для определения местоположения объектов в файле ресурсов пользовательского интерфейса. Когда Вы выбираете данную команду, открывается диалоговое окно Find UIR Objects, представленное на Рисунке 2-18.

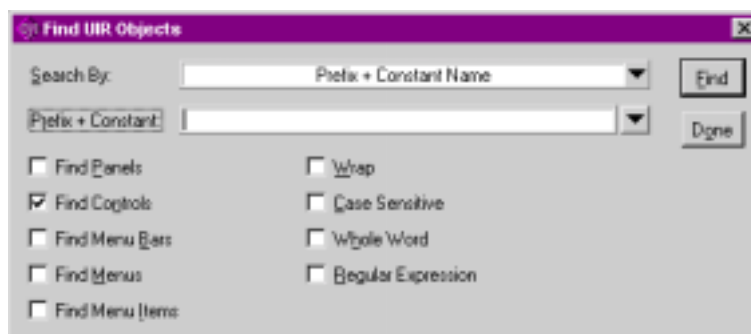


Рисунок 2-18. Диалоговое окно Find UIR Objects.

Выбирайте тип(ы) объектов, поиск которых Вы хотите осуществить, путем проверки соответствующих контрольных окон в левой колонке диалогового окна.

Критерий поиска выбирается из элемента управления кольцевым списком **Search By**. Предлагаются следующие критерии поиска:

- Постоянный префикс (достоверный критерий для панелей и линеек меню)
- Постоянное имя (достоверный критерий для элементов управления, меню и элементов меню)
- Префикс + Постоянное имя (достоверен для всего)
- Имя функции Callback (недостоверен только для линеек меню)
- Метка (недостоверен только для линеек меню)

Введите текст, поиск которого Вы хотите осуществить, в строчный элемент управления. Вы можете просмотреть список всех строк в файле, которые соответствуют текущему критерию поиска, нажатием стрелки справа от строчного элемента управления или с помощью клавиш со стрелками, направленными вверх и вниз.

При достижении конца файла, для продолжения поиска с начала файла, используйте **Wrap**.

Опция **Case Sensitive** обнаруживает только те экземпляры текста, которые полностью соответствуют заданному.

Если Вы выбираете **Regular Expression**, LabWindows/CVI воспринимает определенные символы в осуществляющем поиск строчном элементе управления как символы регулярного выражения, а не как буквенные символы.

Для осуществления поиска нажмите кнопку **Find**. Если какие-либо объекты пользовательского интерфейса соответствуют заданному критерию, то диалоговое окно заменяется на окно, приведенное на Рисунке 2-19.

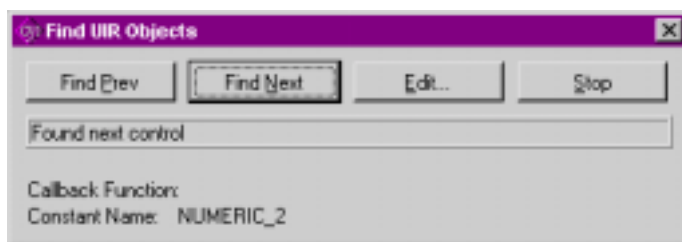


Рисунок 2-19. Диалоговое окно Find UIR Objects после осуществления поиска.

Данное диалоговое окно позволяет Вам просмотреть список найденных соответствующих вариантов. При переходе к каждому новому объекту списка в окне появляются метка и имя его функции обратного вызова, а также высвечивается сам объект.

Find Prev осуществляет поиск предыдущего соответствующего объекта.

Find Next осуществляет поиск следующего соответствующего объекта.

Команда **Edit** завершает поиск и выводит диалоговое окно **Edit** для высвеченного в данный момент объекта пользовательского интерфейса.

Команда **Stop** завершает поиск.

Show/Hide Panels

Команда **Show/Hide Panels** имеет подменю, представленное на Рисунке 2-20.

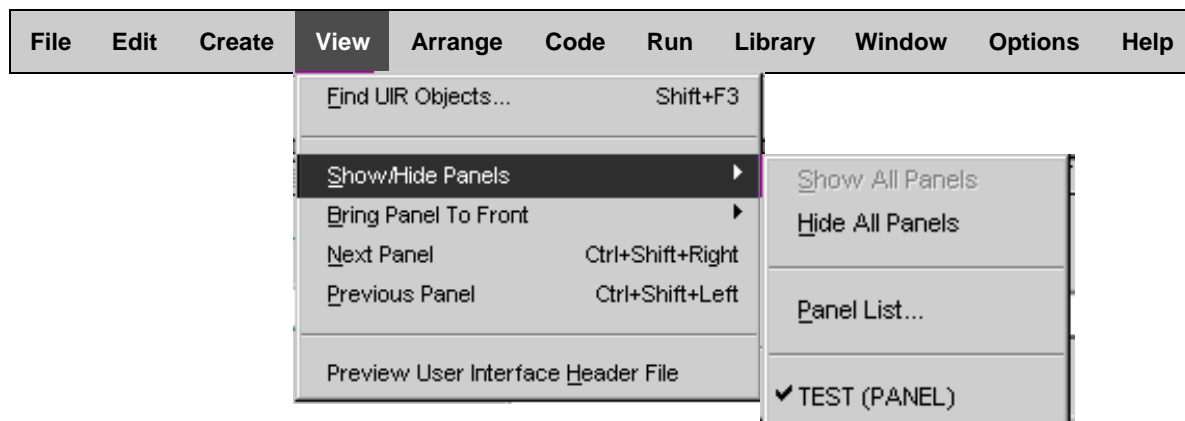


Рисунок 2-20. Подменю Show/Hide Panels.

Используйте подменю для выбора отдельных панелей для просмотра в Редакторе пользовательского интерфейса или для выбора **Show All Panels** или **Hide All Panels**.

Команда **Bring Panel to Front** имеет подменю, позволяет Вам выбрать панель из списка и вынести ее на передний план для редактирования.

Команда **Next Panel** выносит следующую панель в файле `.uir` на передний план для редактирования.

Команда **Previous Panel** выносит предыдущую панель в файле `.uir` на передний план для редактирования.

Preview User Interface Header File

Команда **Preview User Interface Header File** открывает окно Исходного кода с предварительным просмотром файла заголовков, который, при условии его сохранения, будет связан с файлом `.uir` в окне Редактора пользовательского интерфейса.

Меню Arrange

В настоящем разделе дано объяснение, как использовать команды меню **Arrange** в окне Редактора пользовательского интерфейса. На Рисунке 2-21 приведено меню **Arrange**.

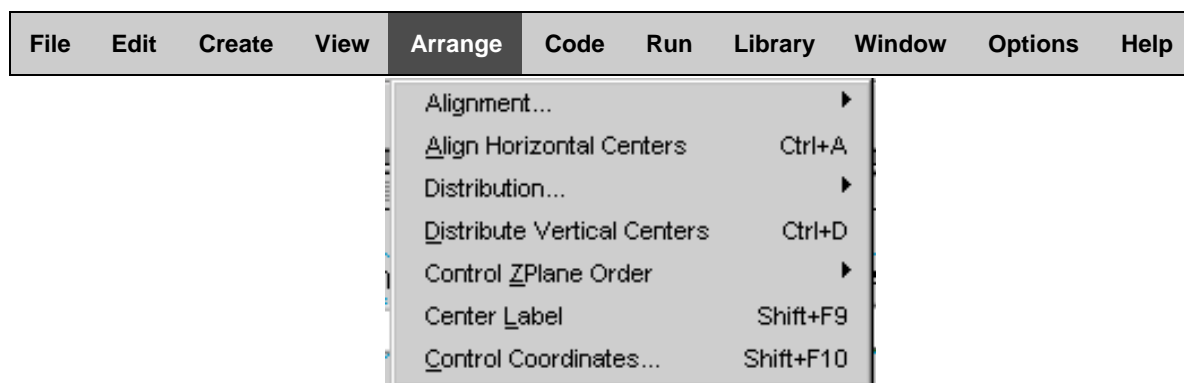


Рисунок 2-21. Меню Arrange.

Alignment...

Команда **Alignment** позволяет упорядочить расположение элементов управления на панели. Для выбора группы элементов воспользуйтесь мышкой, чтобы выделить их и переместить в нужное место, или же Вы можете нажать `<Shift-Click>` на каждый элемент, который требуется включить в группу. После этого, Вы можете выбрать метод упорядочения из подменю, приведенного на Рисунке 2-22.

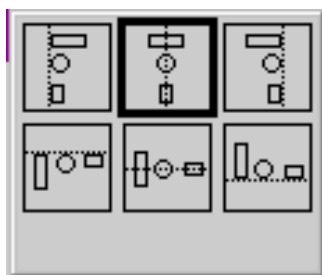


Рисунок 2-22. Меню Alignment.



Left Edges (По левому краю) осуществляет вертикальное упорядочение выбранных элементов управления по левому краю самого крайнего элемента управления слева.



Horizontal Centers (Горизонтально по центру) вертикально упорядочивает выбранные элементы управления по их горизонтальному центру.



Right Edges (По правому краю) осуществляет вертикальное упорядочение выбранных элементов управления по правому краю самого крайнего элемента управления справа.



Top Edges (По верхнему краю) осуществляет горизонтальное упорядочение выбранных элементов управления по верхнему краю самого верхнего элемента управления.



Vertical Centers (Вертикально по центру) осуществляет горизонтальное упорядочение выбранных элементов управления относительно их вертикального центра.



Bottom Edges (По нижнему краю) осуществляет горизонтальное упорядочение выбранных элементов управления по нижнему краю самого нижнего элемента управления.

Align Horizontal Centers

Команда **Align Horizontal Centers** выполняет ту же операцию, что и команда **Alignment**, используя опцию упорядочения, выбранную последней из подменю команд **Alignment**.

Distribution...

Команда **Distribution** позволяет распределять элементы управления на панели. Выберите группу элементов, выделив их мышкой или нажав <Shift-Click> на каждый элемент, который Вы хотите включить в группу. После этого, Вы можете выбрать метод распределения из подменю, показанного на Рисунке 2-23.

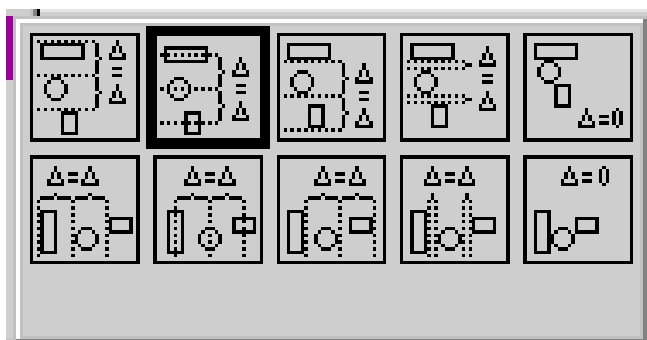


Рисунок 2-23. Подменю Distribution.



Top Edges (По верхнему краю) устанавливает равные вертикальные пробелы между верхними краями элементов управления. Якорными точками служат самый верхний и самый нижний элементы управления.



Vertical Centers (Вертикально по центру) устанавливает равные вертикальные пробелы между центрами элементов управления. Якорными точками служат самый верхний и самый нижний элементы управления.



Bottom Edges (По нижнему краю) устанавливает равные вертикальные пробелы между нижними краями элементов управления. Якорными точками служат самый верхний и самый нижний элементы управления.



Vertical Gap (Вертикальный промежуток) устанавливает равный вертикальный промежуток между элементами управления. Якорными точками служат самый верхний и самый нижний элементы управления.



Vertical Compress (Вертикальное сжатие) сжимает пробелы между элементами управления, чтобы удалить все вертикальные промежутки между ними.



Left Edges (По левому краю) устанавливает равные горизонтальные промежутки между левым краем элементов управления. Якорными точками служат крайний левый и крайний правый элементы управления.



Horizontal Centers (Горизонтально по центру) устанавливает равные горизонтальные промежутки между центрами элементов управления. Якорными точками служат крайний левый и крайний правый элементы управления.



Right Edges (По правому краю) устанавливает равные горизонтальные промежутки между правым краем элементов управления. Якорными точками служат крайний левый и крайний правый элементы управления.



Horizontal Gap (Горизонтальный промежуток) устанавливает равный горизонтальный промежуток между элементами управления. Якорными точками служат крайний левый и крайний правый элементы управления.



Horizontal Compress (Горизонтальное сжатие) сжимает пробелы между элементами управления, чтобы удалить все горизонтальные промежутки между ними.

Distribute Vertical Centers

Команда **Distribute** выполняет ту же операцию, что и команда **Distribution** относительно опции, выбранной последней из подменю команды **Distribution**.

Control ZPlane Order

Опция **Control ZPlane Order** позволяет устанавливать последовательность, в которой будут выводиться элементы управления. Элементы управления всегда выводятся по порядку, от конца к началу порядка z-plane. В меню **Control ZPlane Order** представлены четыре команды:

- **Move Forward** перемещает элемент управления на одно место вперед в порядке (последовательности) z-plane.
- **Move Backward** перемещает элемент управления на одно место назад в порядке (последовательности) z-plane.
- **Move to Front** перемещает элемент управления на первое место в последовательности z-plane, так что он выводится последним.
- **Move to Back** перемещает элемент управления на последнее место в последовательности z-plane, так что он выводится первым.

Center Label

Команда **Center Label** центрует метку выбранного элемента управления.

Control Coordinates...

Команда **Control Coordinates** вызывает диалоговое окно, позволяющее интерактивно устанавливать ширину, высоту, верхнюю и нижнюю границы всех выбранных элементов управления и меток.

Меню Code

Используйте команды в меню **Code** для автоматического генерирования кода на основе файла (.uir), который Вы создаете или редактируете. На Рисунке 2-24 представлено меню **Code**.

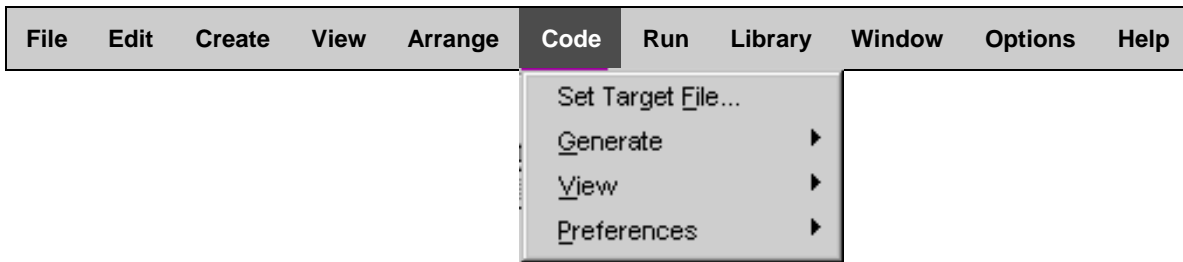


Рисунок 2-24. Меню Code.

Set Target File...

Используйте команду **Set Target File** для задания файла, в который LabWindows/CVI будет генерировать код. Выбор данной команды открывает диалоговое окно Set Target File (Установка целевого файла), показанное на Рисунке 2-25. По умолчанию, LabWindows/CVI устанавливает сгенерированный код в новое окно, если не открыт файл исходного кода (.c). После этого, данный исходный файл становится принимаемым по умолчанию целевым файлом. Построитель кодов использует этот же целевой файл как целевой файл панели функций, за исключением случаев, когда целевым файлом панели функций является окно Интерактивного выполнения.

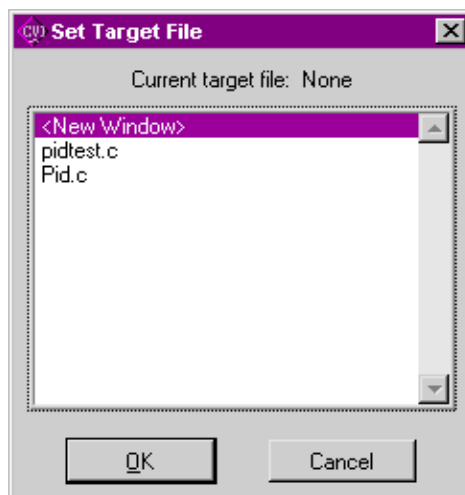


Рисунок 2-25. Диалоговое окно Set Target File (Установка целевого файла).

Для установки целевого файла выберите файл из списка опций в диалоговом окне Set Target File, а затем выберите **ОК**. Опции включают все открытые исходные файлы и новое окно.

Generate

Команды в меню **Generate** создают код на основе файла `.uir`. На Рисунке 2-26 представлено меню **Generate**. Код, созданный меню **Generate**, использует стили в квадратных скобках, заданные с помощью команды **Bracket Styles** в меню **Options** Исходного окна.

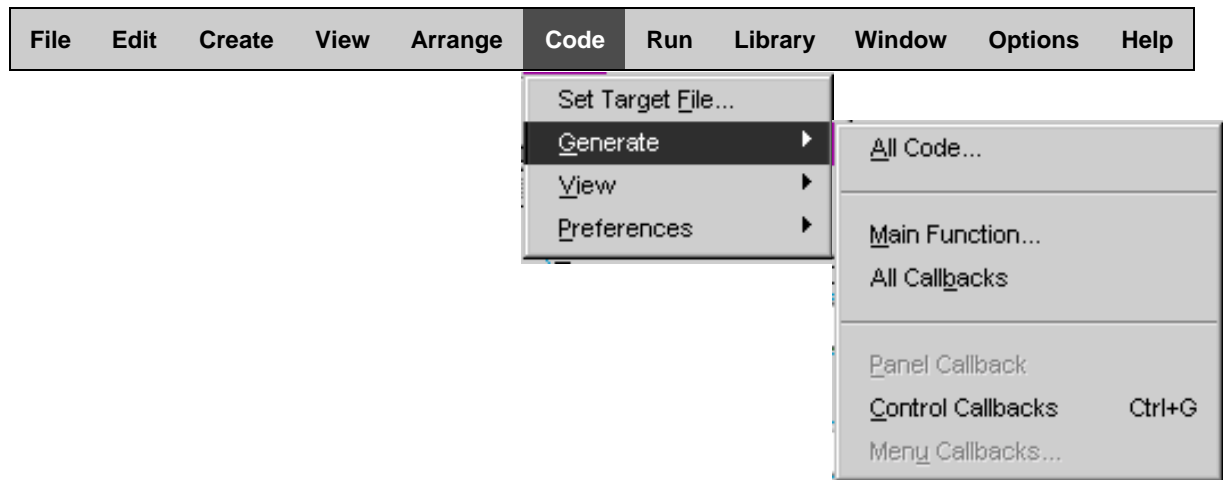


Рисунок 2-26. Меню Generate.

Внимание: *Panel Callback и Control Callback могут вызываться, когда Вы выбираете любую панель или элемент управления, для которых Вы задали имя функции обратного вызова. All Callback может вызываться после того, как Вы создали любой объект с именем функции обратного вызова. Menu Callback может вызываться после того, как Вы создали линейку меню, содержащую, как минимум, один элемент с именем функции обратного вызова.*

Если Вы генерируете код, сопровождающий файл `.uir`, LabWindows/CVI устанавливает скелет кода в целевой файл. Вы должны сохранить файл `.uir` прежде, чем сможете генерировать какой-либо код на основе данного файла. Если Вы сохранили файл `.uir`, LabWindows/CVI сгенерирует заголовочный файл (файл с суффиксом `.h`) под тем же именем. Данный файл `.h` и `userint.h` включаются в исходный файл.

Если Вы попытаетесь сгенерировать одну и ту же функцию более одного раза, то на экране появится диалоговое окно Generate Code (Генерация кода). На Рисунке 2-27 представлено диалоговое окно Generate Code. Каждый только что сгенерированный фрагмент кода высвечивается на экране. Выберите в диалоговом окне Generate Code опцию для замены существующей функции, введения новой функции или перехода к следующей сгенерированной функции.

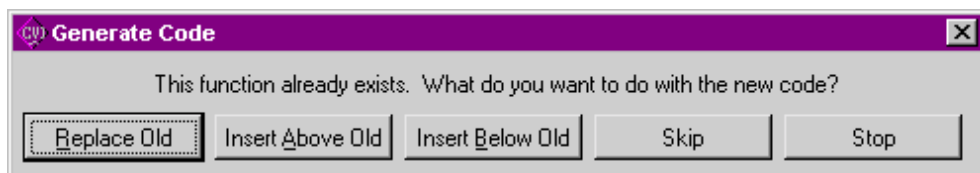


Рисунок 2-27. Диалоговое окно Generate Code.

All Code...

Используйте команду **All Code** для генерирования кода, сопровождающего файл `.uir`. Выбор **Code** » **Generate** » **All Code** откроет диалоговое окно `Generate All Code`, показанное на Рисунке 2-28. Данное диалоговое окно выдает контрольную таблицу и приглашает Вас выбрать панель/панели, загружаемые и выводимые на экран во время работы функцией `main`. `LabWindows/CVI` автоматически задаст принимаемое по умолчанию имя переменной панели для каждой панели в файле `.uir`.

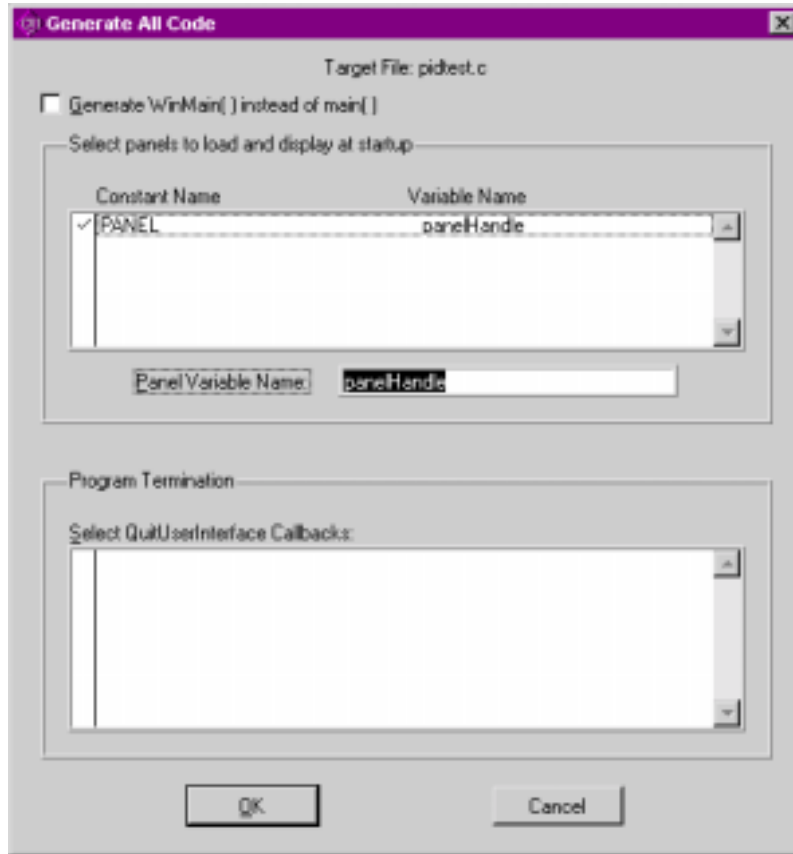


Рисунок 2-28. Диалоговое окно `Generate All Code`.

Кроме того, диалоговое окно `Generate All Code` приглашает Вас выбрать функцию(и) обратного вызова, которая будет завершать программу. Чтобы программа Построителя кодов завершилась успешно, Вы должны включить вызов в функцию `QuitUserInterface`.

Внимание: *Функции обратного вызова, связанные с близкими элементами управления, автоматически проверяются в разделе **Program Termination (Завершение программы)** диалогового окна **Generate All Code**. Вы можете задать близкие элементы управления в диалоговом окне **Edit Panel**.*

Для автоматического генерирования всех кодов выберите панель, которую Вы хотите загрузить и вывести на дисплей пользовательского интерфейса. Кроме того, выберите функцию(и) обратного вызова, которая будет завершать программу, а затем выберите **ОК**.

Когда Вы выбираете **Code** » **Generate** » **All Code**, `LabWindows/CVI` создает операторы `#include`, декларации переменных, скелеты функций и функцию `main` и устанавливает их в целевой файл. Выбранная Вами для выполнения завершения программы функция обратного вызова включает вызов функции `QuitUserInterface` Библиотеки пользовательского интерфейса.

Если Вы не выбрали опцию **Always Append Code to End**, `LabWindows/CVI` устанавливает скелет программы для каждой функции обратного вызова в место, где расположен курсор в целевом файле.

Если курсор находится внутри существующей функции, то, перед вводом новой функции, LabWindows/CVI переносит его в конец существующей функции. Построитель кодов вводит все функции одного типа (панели, элементы управления или меню) вместе в исходный файл. Сначала, в исходный файл устанавливаются любые обратные вызовы панелей, затем устанавливаются обратные вызовы элементов управления и, в последнюю очередь, обратные вызовы меню. Дополнительную информацию о задании местоположения сгенерированного кода смотрите в разделе *Preferences (Предпочтения/Привилегии)* в данной главе.

Скелеты функций для обратных вызовов элементов управления и панелей включают полный прототип, собственный синтаксис, возвращаемое значение и конструкцию “switch”, содержащую case для каждого принимаемого по умолчанию события элемента управления или панели. Скелеты функций для обратных вызовов меню включают полный прототип, а также открывающие и закрывающие квадратные скобки. Вы можете установить принимаемые по умолчанию события, выбрав **Code » Preferences**. Дополнительную информацию смотрите в разделе *Preferences*. Вы можете установить местоположение открывающих и закрывающих квадратных скобок, выбрав команду **Bracket Style** из меню **Options** Исходного окна.

Main Function...

Используйте команду **Main Function** для генерирования кода функции main и записи его в целевой файл. Выбор опции **Code » Generate » Main Function** открывает диалоговое окно Generate Main Function (Генерирование функции main), представленное на Рисунке 2-29. Данное диалоговое окно приглашает Вас выбрать панель, в которую будет загружаться и выдаваться во время работы функция main. LabWindows/CVI автоматически задаст принимаемое по умолчанию имя переменной панели для каждой панели в файле .uir.

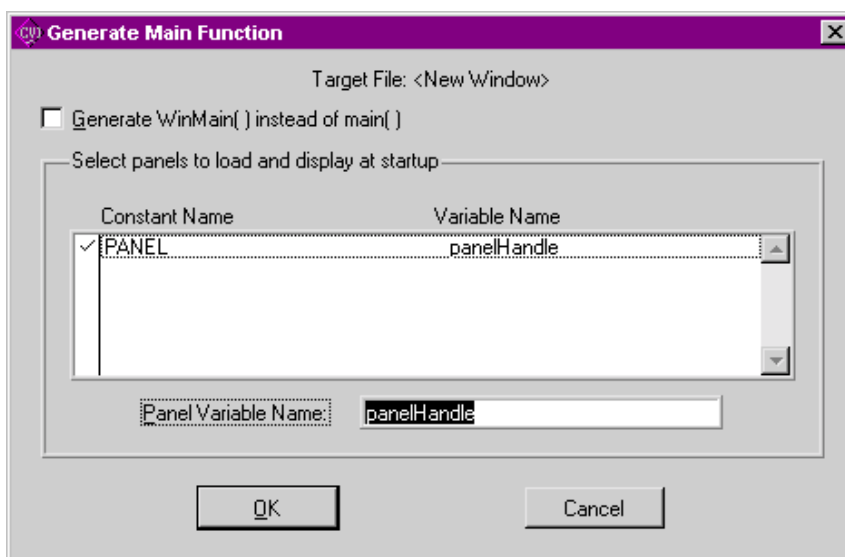


Рисунок 2-29. Диалоговое окно Generate Main Function.

Внимание: Если Вы ранее выбрали команду **Code » Generate » All Code**, то нет необходимости выполнять эту команду. Используйте данную команду, только если Вы хотите заменить функцию обратного вызова main, чтобы добавить или изменить панели, загружаемые во время работы.

Для автоматической генерации кода функции main необходимо выбрать панель(и), которые Вы хотите загрузить и выводить на дисплей пользовательского интерфейса, а затем выбрать **ОК**.

Если Вы выбираете **Code » Generate » Main Function**, LabWindows/CVI создает операторы `#include`, декларации переменных, функцию обратного вызова `main` и устанавливает их в целевой файл.

Внимание: *Если исходный файл содержит только функцию `main` и операторы `#include`, а Вы не создали еще соответствующие функции обратного вызова, то при попытке запуска выполнения проекта Вы можете получить сообщение об ошибке. Если функция `main` вызывает `LoadPanel`, LabWindows/CVI будет генерировать нефатальную ошибку для каждой функции обратного вызова, которую она не сможет найти в исходном файле.*

All Callbacks

Используйте команду **All Callbacks** для генерирования кода для всех функций обратного вызова и записи их в целевой файл.

Если Вы выбираете **Code » Generate » All Callbacks**, LabWindows/CVI создает операторы `#include` и скелеты функций обратного вызова и устанавливает их в целевой файл.

Panel Callback

Используйте команду **Panel Callback** для генерирования кода панели, соединенной с функцией обратного вызова. Для того, чтобы выбрать **Code » Generate » Panel Callback**, необходимо сначала активизировать панель.

Если Вы выбираете **Code » Generate » Panel Callback**, LabWindows/CVI создает операторы `#include` и скелет функции и устанавливает их в целевой файл.

Control Callbacks

Используйте команду **Control Callbacks** для генерирования кода элемента управления, связанного с функцией обратного вызова. Для того, чтобы выбрать **Generate » Control**, Вы сначала должны выбрать, как минимум, один элемент управления.

Если Вы выбираете **Code » Generate » Control Callbacks**, LabWindows/CVI создает операторы `#include` и скелет функции для каждого выбранного элемента управления и устанавливает их в целевой файл.

Скелет функции обратного вызова элемента управления можно также генерировать, нажав на элемент управления правой кнопкой мышки и выбрав из всплывающего меню команду **Generate Control Callback**.

Menu Callbacks...

Используйте команду **Menu Callbacks** для генерирования кода меню и элементов меню, связанных с функциями обратного вызова.

Выбор **Code » Generate » Menu Callbacks** открывает диалоговое окно **Select Menu Bar Objects** (Выбор объектов линейки меню). Вы должны выбрать объекты линейки меню, для которых Вы хотите сгенерировать обратные вызовы, а затем выбрать **OK**.

Когда Вы выбираете **OK**, LabWindows/CVI создает операторы `#include`, прототипы функций, а также открывающие и закрывающие квадратные скобки для каждой функции обратного вызова. Конструкции `switch` и операторы `case` не создаются, так как обычные принимаемые по умолчанию операторы не применяются к функциям **Menu Callback**. Вы должны добавить код для реализации операций, которые, по Вашему желанию, будут выполняться при выборе элемента линейки меню.

View

Используйте команду **View** для просмотра кода для данной функции обратного вызова. На Рисунке 2-30 представлено меню **View**.

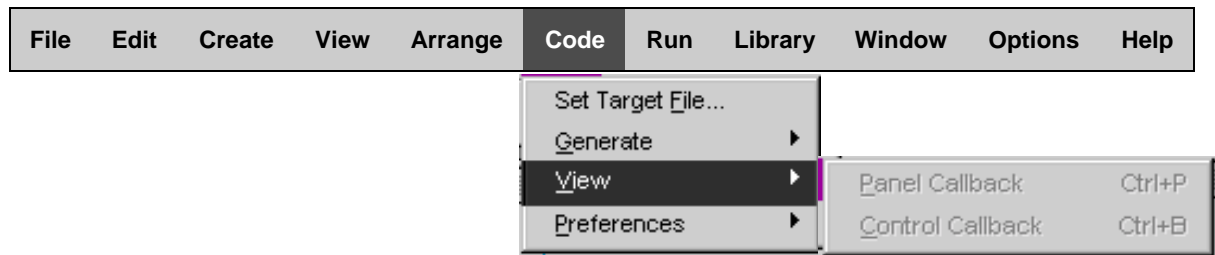


Рисунок 2-30. Меню View.

Для просмотра кода функции из файла `.uir` нужно выбрать панель или элемент управления, а затем выбрать **View** » **Panel Callback** или **View** » **Control Callback**. Исходный файл, содержащий функцию обратного вызова, появляется на экране с высвеченным именем функции. Вы также можете просмотреть код функции обратного вызова элемента управления, нажав на элемент управления правой кнопкой мышки и выбрав из всплывающего меню команду **View Control Callback**.

Если Вы выбираете команду **View** для функции обратного вызова, LabWindows/CVI осуществляет поиск данной функции во всех открытых Исходных окнах и во всех исходных файлах проекта, а также во всех остальных открытых исходных файлах. Если функция обнаружена в закрытом файле проекта, то этот файл открывается автоматически.

Команда **View** полезна потому, что функции обратного вызова для одного пользовательского интерфейса могут находиться в нескольких различных файлах и прокрутка исходного кода не будет эффективной. С помощью команды **View** Вы можете мгновенно перейти из файла пользовательского интерфейса к функции обратного вызова объекта, независимо от того, открыт или закрыт исходный файл.

Закончив просмотр кода, Вы можете мгновенно вернуться из исходного файла в файл `.uir`. Для того, чтобы вернуться в файл `.uir`, установите курсор на имя функции обратного вызова или на постоянное имя объекта Пользовательского интерфейса, в который Вы хотите перейти, после чего выберите команду **Find UI Object** из меню **View** в Исходном окне.

Внимание: *Нельзя использовать команду View для функций обратного вызова меню.*

Preferences

Используйте команду **Preferences** (Параметры) для изменения принимаемых по умолчанию установок для операторов case, сгенерированных для функций обратного вызова элементов управления, или для функций обратного вызова панелей, или для задания местоположения целевого файла для сгенерированного кода. На Рисунке 2-31 представлено подменю **Preferences**.

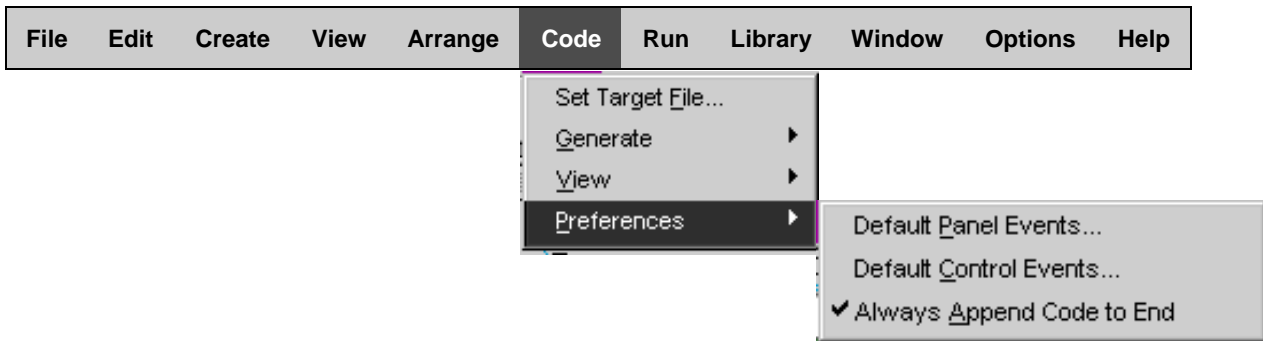


Рисунок 2-31. Меню Preferences.

Default Panel Events... и Default Control Events...

Используйте команды **Default Panel Events** и **Default Control Events** для выбора, какие события LabWindows/CVI будет устанавливать в конструкцию switch кода для функций обратного вызова панели или элемента управления, соответственно. Вы можете выбирать из нескольких событий, а также Вы можете добавить “default:” (по умолчанию) switch case. Выбор **Code** » **Preferences** » **Default Panel Events** откроет диалоговое окно Panel Callback Events (События обратного вызова панели). Выбор **Code** » **Preferences** » **Default Control Events** откроет диалоговое окно Control Callback Events (События обратного вызова элементов управления).

Для установки **Default Panel Events** или **Default Control Events** необходимо выбрать события, которые Вы хотите включить в код в качестве операторов case, а затем выбрать **ОК**. Для каждой выбранной Вами опции LabWindows/CVI включает в исходный код оператор case, соответствующий данной опции.

Внимание: *Принимаемые по умолчанию события элементов управления игнорируются для Обратных вызовов Таймера, так как они реагируют только на следующие case событий: EVENT_TIMER_TICK и EVENT_DISCARD.*

Always Append Code to End

При выборе данной опции LabWindows/CVI устанавливает скелет кода для каждой функции обратного вызова в конце целевого файла. Если данная опция не задана, то вновь сгенерированный код будет устанавливаться в текущее положение курсора в целевом файле.

Меню Run

Меню **Run** содержит набор команд, появляющихся в меню Run исходного окна. В данный набор входят следующие команды:

Run Project
Continue
Step Over

Step Into
Finish Function
Terminate Execution

Break at First Statement
Breakpoints

Описание каждой из этих команд можно найти в разделе *Меню Run*, в Главе 4, *The Source, Interactive Execution, and Standard Input/Output Windows (Окна Интерактивного выполнения, Стандартного ввода-вывода и исходное)*, в *Справочном руководстве пользователя LabWindows/CVI*.

Меню Library

Меню **Library** в окнах Редактора пользовательского интерфейса работает так же, как и меню **Library** в окне Project. Информация о меню **Library** представлена в Главе 3, *Окно Project*, в *Справочном руководстве пользователя LabWindows/CVI*.

Меню Window

Меню **Window** в окнах Редактора пользовательского интерфейса работает так же, как и меню **Window** в окне Project. Информация о меню **Window** представлена в Главе 3, *Окно Project*, в *Справочном руководстве пользователя LabWindows/CVI*.


Меню Options

В данном разделе описано использование команд меню **Options**. На Рисунке 2-32 представлено меню **Options**.






Рисунок 2-32. Меню Options.

Operate Visible Panels

Operate Visible Panels позволяет работать с видимыми панелями так же, как Вы делаете это в прикладной программе. Данная команда выполняет то же действие, что и нажатие на иконку . Чтобы вернуться в режим редактирования после окончания работы с панелью, нужно еще раз выбрать **Operate Visible Panels**.

Next Tool

Команда **Next Tool** в меню **Options** циклически прокручивает в Редакторе пользовательского интерфейса три или четыре его режима: режим редактирования , режим установки меток  и режим задания цвета .

Preferences...

Выбор **Preferences** (Параметры) открывает диалоговое окно Preferences Редактора пользовательского интерфейса, представленное на приведенном ниже рисунке.

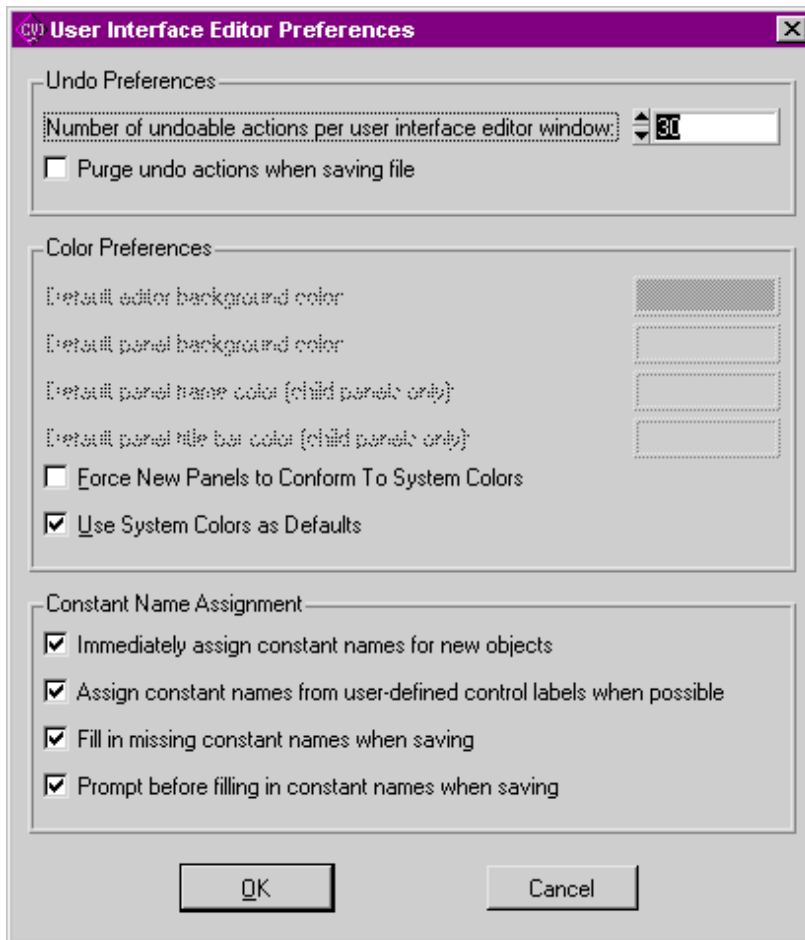


Рисунок 2-33. Диалоговое окно User Interface Preferences (Параметры пользовательского интерфейса).

Undo Preferences

Используйте раздел Undo Preferences (Параметры возврата) диалогового окна Preferences Редактора пользовательского интерфейса для установки ряда операций возврата для каждого файла. Если Вы хотите, чтобы информация в буфере Undo стиралась при каждом сохранении файла, выберите независимый переключатель **Purge undo actions when saving file**.

Color Preferences

Используйте раздел Color Preferences (Параметры цвета) диалогового окна Preferences Редактора пользовательского интерфейса для установки принимаемых по умолчанию цветов редактора и панели.

Constant Name Assignment

Постоянные имена связывают объекты GUI пользователя и Вашу программу. Редактор пользовательского интерфейса записывает все присвоенные постоянные имена в заголовочный файл, соответствующий загруженному файлу `.uir`.

Раздел Constant Name Assignment (Присвоение постоянных имен) диалогового окна Preferences Редактора пользовательского интерфейса позволяет Вам установить параметры присвоения постоянных имен в случае, когда Вы не присваиваете постоянные имена сами.

При вызове опции “Immediately assign constant names for new objects” (“Немедленное задание постоянных имен для новых объектов”), Редактор пользовательского интерфейса будет генерировать постоянные имена для каждого создаваемого Вами объекта. Для панелей и элементов управления сгенерированное постоянное имя возникает в диалоговом окне редактирования при первом его вызове. Для линеек меню постоянные имена присваиваются только в том случае, если Вы выходите из редактора линеек меню. Во всех случаях Вы можете свободно модифицировать сгенерированные постоянные имена.

Рекомендуется оставлять активной опцию “Immediately assign constant names for new objects”. Это упростит для Вас использование свойств LabWindows/CVI, которые были разработаны, чтобы помочь Вам в написании программы по работе с пользовательским интерфейсом.

Помните, что, когда активна опция “Immediately assign constant names for new objects”, опция “Assign constant names from user-defined control labels when possible” (“Присвоение, где возможно, постоянных имен из определенных пользователем меток элементов управления”) не может быть вызвана. Это объясняется тем, что принимаемые по умолчанию метки элементов управления “Untitled Control” (“Неозаглавленный элемент управления”) не были заменены на другие во время генерирования постоянного имени. Вместо этого, постоянное имя будет основываться на типе элемента управления.

Если Вы решите отключить опцию опции “Immediately assign constant names for new objects”, то рекомендуется активизировать опцию “Fill in missing constant names when saving” (“Вставить недостающие постоянные имена при сохранении”).

Assign Missing Constants...

Команда **Assign Missing Constants** (Присвоить недостающие постоянные имена) присваивает постоянные имена всем объектам в окне Редактора пользовательского интерфейса, которые в данный момент не имеют постоянных имен. Появляется диалоговое окно подтверждения, показывающее количество элементов, не имеющих постоянных имен.

Save in Text Format

Команда **Save in Text Format** сохраняет содержание окна Редактора пользовательского интерфейса в текстовом формате ASCII. Появляется диалоговое окно, приглашающее Вас ввести имя пути, под которым будет сохранен текстовый файл. Для таких файлов рекомендуется расширение `.tui`. НЕЛЬЗЯ использовать расширение `.uir`.

Текстовый файл ASCII содержит описания всех объектов в окне Редактора пользовательского интерфейса.

Внимание: *Если в Вашем Окне Редактора пользовательского интерфейса находится много объектов, то для выполнения этой команды может потребоваться довольно много времени (например, 3 минуты).*

Load From Text Format...

Команда **Load From Text Format** (Загрузить из текстового формата) загружает в новое окно Редактора пользовательского интерфейса объекты, определенные в файле, сохраненном с помощью команды Save in Text Format. Появляется диалоговое окно, запрашивающее у Вас имя пути файла.