

Глава 1

Введение

В этой главе мы расскажем о библиотеке wxWidgets и основных областях ее использования: коснемся истории проекта, вопросов лицензирования, архитектуры библиотеки и использующих эту библиотеку подпроектов (портов), а также расскажем о сообществе пользователей данной библиотеки.

1.1 Что такое wxWidgets?

wxWidgets — это инструментарий для написания десктопных и мобильных приложений с графическим пользовательским интерфейсом (graphical user interface, GUI). Эта библиотека является фреймворком, т.е. она реализует большую часть работы и поведения окон. Библиотека wxWidgets содержит огромное число классов и методов, которые программист может использовать и настраивать. Приложение обычно использует окно, содержащее стандартные элементы управления, возможно показывающее специальные изображения или графику, а также реагирующее на ввод с клавиатуры, манипулятора мыши или из других источников. Приложение также может взаимодействовать с другими процессами или управлять другими программами. Таким образом, wxWidgets позволяет легко написать приложения, которые могут делать все то, что могут делать современные программы.

Несмотря на то, что wxWidgets часто называют «инструментарием для разработки GUI», фактически библиотека позволяет сделать намного больше и может быть полезна в самых разных областях разработки. Это следствие того, что приложения wxWidgets должны иметь возможность портирования на различные платформы, а поэтому библиотека содержит классы для работы с файлами и потоками ввода/вывода, нитями, настройками приложения, межпроцессорной коммуникацией, файлами помощи, базами данных и многим другим.

1.2 Почему стоит использовать wxWidgets?

Одной из особенностей, выгодно отличающих wxWidgets от других фреймворков, таких как MFC или OWL, является ее поразительная переносимость. У wxWidgets есть программный интерфейс для приложений (Application Programming Interface, API), который одинаков (или почти одинаков) на всех поддерживаемых платформах. Это означает, к примеру, что вы можете написать свое приложение в системе

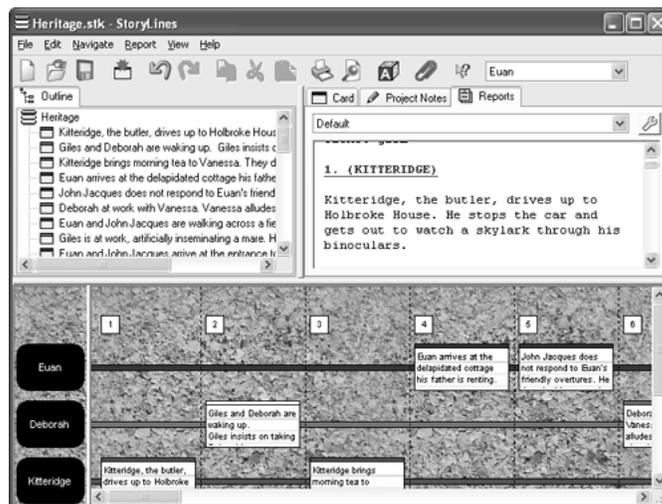


Рис. 1.1: StoryLines в системе Windows

Windows и, сделав небольшие изменения (или даже обойдясь без них), перекомпилировать вашу программу в Linux или Mac OS X. Это обстоятельство дает огромное преимущество по сравнению с полным переписыванием приложения под каждую платформу и в том числе означает, что вам не нужно изучать различные API для каждой платформы. Кроме того, библиотека позволяет идти в ногу с прогрессом. Когда меняется программная среда, wxWidgets меняется вместе с ней, позволяя вашим приложениям портироваться на самые последние версии систем, поддерживая большую часть их новых возможностей.

Другой приятной особенностью является то, что wxWidgets реализует родной вид и поведение программ. Некоторые фреймворки используют один и тот же код для элементов управления на различных платформах (возможно используя специальные темы, чтобы элементы управления были похожи на родные). В отличие от них, wxWidgets использует родные элементы везде, где это только возможно (и свой собственный набор элементов в противном случае), поэтому элементы не просто похожи на родные элементы платформы — они ими и являются. Это очень важно для конечного пользователя, так как даже самая маленькая, почти незаметная разница в поведении программы, по сравнению со стандартом платформы, может создать ощущение некоторой искусственности программы. Для примера рассмотрим рис. 1.1, показывающий wxWidgets-приложение StoryLines (это инструмент, помогающий писателям графически представить хронологию произведения), запущенное в системе Windows XP.

Этот вид полностью соответствует виду типичного Windows-приложения с вкладками, полосами прокрутки и выпадающим списком, соответствующими выбранной пользователем теме. Аналогично, рис. 1.2 показывает приложение StoryLines, запущенное под Mac OS X с ожидаемым для Aqua поведением и видом. В приложении отсутствует строка меню, так как, следуя соглашениям, принятым в Mac OS, меню для текущего окна отображается в верхней строке экрана.

Наконец, рис. 1.3 показывает StoryLines как приложение GTK+, запущенное в системе Red Hat Linux.

Почему бы не использовать для этого Java? Несмотря на то, что Java великолепно

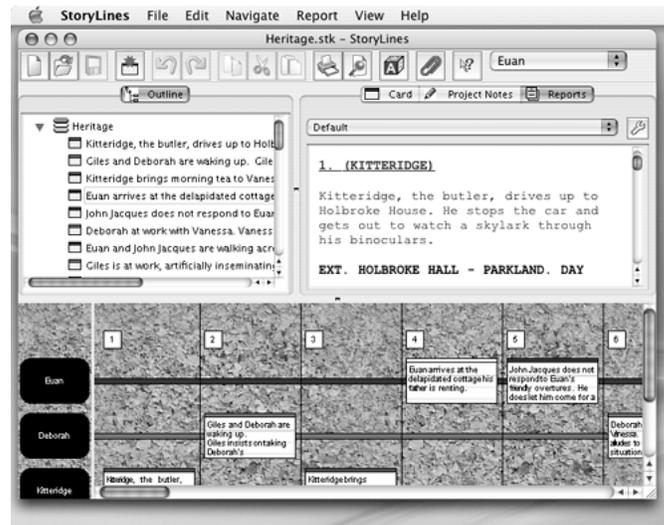


Рис. 1.2: StoryLines в системе Mac OS X

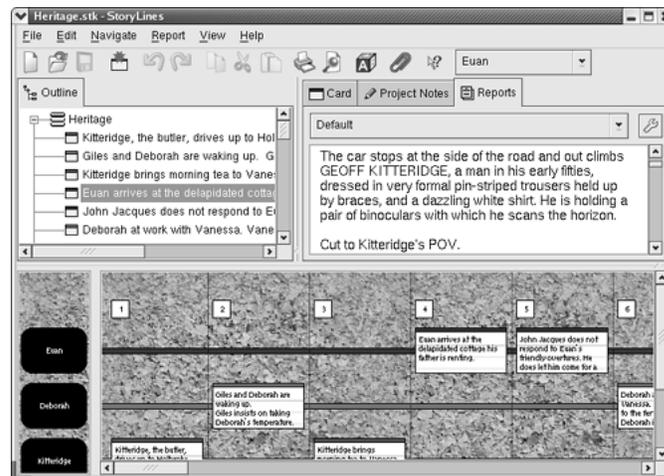


Рис. 1.3: StoryLines в системе Linux

подходит для web-приложений, она не всегда является хорошим выбором для десктопа. В частности, приложения на C++, использующие wxWidgets, быстрее, имеют более естественный вид и предсказуемое поведение, а также проще в установке, так как не зависят от капризной виртуальной машины Java. Использование C++ также позволяет получить доступ к низкоуровневой функциональности и легче интегрировать свое приложение с существующим кодом на C и C++. Именно по этим причинам очень мало популярных прикладных программ базируется на Java. Таким образом, wxWidgets позволит вам писать быстрые приложения, которые будут понятны пользователю.

wxWidgets является проектом с открытым исходным кодом. Обычно это означает, что вам не нужно платить за его использование (пока вы не захотите сделать пожертвование проекту!), но также имеет очень большое философское и стратегическое значение. Продукты с открытым кодом всегда противопоставляют их проприетарным эквивалентам. Используя wxWidgets, вы всегда знаете, что код, на который вы полагаетесь, никогда не исчезнет. Вы всегда сможете самостоятельно исправить любую проблему, поправив исходный код. Можно получить большое удовольствие, принимая участие в обсуждении проекта, нежели общаясь с персоналом, осуществляющим корпоративную поддержку проприетарных продуктов. Участники открытого проекта находятся там, так как они любят то, что делают и с удовольствием делятся своими знаниями, тогда как члены корпоративной поддержки не всегда так идеалистически мотивированы. Когда вы используете wxWidgets, вы попадаете во впечатляющее сообщество талантов, обладающих большими знаниями. Многие аспекты разработки приложений, которые в ином случае пришлось бы придумывать самостоятельно, инкапсулированы этими разработчиками в легко используемые классы, которые вы можете подключить к своему коду. Активное пользовательское сообщество проконсультирует вас в случае необходимости в списках рассылки. В рассылке можно обсудить не только wxWidgets, но также и другие вещи, близкие сердцу любого разработчика. Возможно, в один прекрасный день вы сами присоединитесь к wxWidgets и станете ее активным разработчиком.

Библиотека wxWidgets широко поддерживается компьютерной индустрией. Список пользователей включает в себя такие известные компании как AOL, AMD, CALTECH, Lockheed Martin, NASA, Open Source Applications Foundation, Xerox и многие другие. wxWidgets объединяет огромное число пользователей: от программистов-одиночек до огромных корпораций, от отделений компьютерных наук до центров медицинских исследований и от экологических организаций до индустрии телекоммуникаций. Библиотека используется множеством открытых проектов, таких как редактор музыки Audacity и системой управления базами данных pgAdmin III.

Люди используют wxWidgets по различным причинам: для одних это простая и элегантная замена MFC, для других — возможность легко менять платформу и перейти, например, с Microsoft Windows на Unix или Mac OS X. Проект wxWidgets также уделяет внимание поддержке мобильных платформ. Существуют порты для встраиваемых систем на базе Linux, Microsoft Pocket PC и (возможно порт скоро появится) Palm OS.

1.3 Краткая история wxWidgets

Проект wxWidgets берет свое начало в 1992, когда Джулиан Сمارт (Julian Smart) начал работать в университете Эдинбурга над разработкой инструмента для создания диаграмм, который назывался Hardy. Ему не хотелось выбирать платформу для работы программы (рабочие станции SUN и PC), поэтому он решил использовать мультиплатформенный фреймворк. В связи с тем, что в то время число существующих фреймворков было невелико, да и покупка одного из них не входила в планы подразделения, было принято решение написать свой собственный. В сентябре 1992 года с разрешения университета на ftp-сервер факультета была помещена первая версия библиотеки, и другие разработчики получили возможность использовать этот проект. В самом начале wxWidgets был написан с использованием XView и MFC 1.0, но так как Borland C++ не поддерживал MFC, то в скором времени библиотека была переписана на чистом Win32. XView берет свое начало из Motif, так что очень скоро появился порт под Motif.

Спустя некоторое время сформировалось маленькое, но весьма активное сообщество пользователей, и был создан почтовый список рассылки. Стали приходить дополнения и исправления ошибок, включая порт на Xt от Маркуса Хользема (Markus Holzem). Проект постепенно привлекал все больше и больше пользователей по всему миру: отдельные пользователи, академики, правительственные подразделения и множество корпоративных пользователей, которые поняли, что wxWidgets предлагает более удачный продукт и поддержку, чем те коммерческие проекты, услугами которых они пользовались или собирались пользоваться.

В 1997 году с помощью Маркуса Хользема был спроектирован новый API для wxWidgets версии 2. Вольфрам Глогер (Wolfram Gloger) решил, что wxWidgets должна быть портирована на GTK+ и адаптировал библиотеку под среду GNOME. Роберт Роублинг (Robert Roebling) стал главным разработчиком для порта wxGTK, который в данный момент является основным для систем Unix/Linux. В 1998 году версии под Windows и GTK+ были объединены и помещены в единое хранилище версий. Также в этом году к проекту присоединился Вадим Зейтлин (Vadim Zeitlin), который добавил большое количество кода и технических решений, а также Стефан Ксомор (Stefan Csomor), начавший портирование библиотеки на платформу Mac OS.

В 1999 увидел свет впечатляющий компонент Вацлава Славика (Vaclav Slavik) — wxHTML и просмотрщик для файлов помощи на базе HTML. В 2000 году компания SciTech Inc. спонсировала начальную разработку wxUniversal — собственных элементов управления wxWidgets, которые можно использовать на платформах, где вообще отсутствуют элементы управления. wxUniversal был впервые использован в приложении SciTech на платформе MGL, которая является низкоуровневой графической прослойкой.

В 2002 году Джулиан Смарт (Julian Smart) и Роберт Роублинг добавили порт wxX11, используя наработки wxUniversal. Требуемый только Unix и X11, порт wxX11 очень хорошо подходит для систем с низкой производительностью.

В июле 2003 wxWidgets смогли запустить на Windows CE, и Роберт Роублинг продемонстрировал приложение wxGTK, запущенное на встроенной Linux-платформе GPE.

В 2004 году под давлением фирмы Microsoft название проекта было изменено с начального «wxWindows» на «wxWidgets».

Также в 2004 Стефан Ксомор и еще часть разработчиков полностью переписали порт wxMac под систему OS X, существенно повысив качество и функциональность приложений в OS X. Под руководством Дэвида Эллиота (David Elliot) порт, использующий Cocoa, также был существенно улучшен. Вильям Осборн (William Osborne) выиграл соревнование по портированию wxWidgets на платформу Palm OS 6 для запуска простейшего примера «minimal» из поставки wxWidgets. Версия 2.6 была выпущена в апреле 2005 года, объединив все основные улучшения во все порты.

Планы проекта на будущее:

- создание инструментов для управления пакетами, которые облегчат интеграцию со сторонними компонентами
- расширение поддержки встраиваемых приложений
- альтернативный механизм обработки сообщений
- добавление сложных компонентов, таких как объединенное со списком дерево
- вторая версия wxHTML с полной поддержкой web на всех платформах
- дальнейшая интеграция со стандартными решениями, такими как STL
- полностью рабочий порт для Palm OS

1.4 Сообщество wxWidgets

Сообщество wxWidgets поддерживает два списка рассылки: wx-users (для пользователей) и wx-dev (для разработчиков). Web-сайт проекта анонсирует новости, содержит документацию и ссылки на релизы, а также Wiki (систему, представляющую из себя множество страниц, которые каждый пользователь может редактировать и дополнять). Также для разработчиков и пользователей доступен форум. Ниже дан список адресов этих ресурсов:

- <http://www.wxwidgets.org> : домашняя страница проекта
- <http://lists.wxwidgets.org> : архивы списков рассылки
- <http://wiki.wxwidgets.org> : система Wiki проекта
- <http://www.wxforum.org> : форум wxWidgets

Как и большинство проектов с открытыми исходниками, wxWidgets разрабатывается с помощью репозитория CVS — системы управления версиями, которая отслеживает изменения в коде. В целях избежания хаоса в разработке, только небольшое число разработчиков могут помещать код непосредственно в CVS. Остальные могут дополнять или исправлять ошибки в коде, публикуя информацию для изменения в специальной системе (в данный момент для этого используется система для обратной связи SourceForge'a). Разработка осуществляется в рамках двух веток: «стабильной» ветки, где правятся только бинарно-совместимые ошибки, и «разрабатываемой» ветки (текущее состояние CVS). В соответствии со статусом версии, стабильные релизы

нумеруются четными числами (например, 2.4.x), а разрабатываемые — нечетными (к примеру, 2.5.x). Пользователи могут ждать новые релизы или скачать исходники из соответствующей ветки CVS.

Решения об изменении в API и другие технические детали принимаются в ходе обсуждения в списке рассылки wx-dev. У некоторых проектов на базе wxWidgets существуют и развиваются свои собственные сообщества пользователей, как, например, у wxPython и wxPerl (смотрите Приложение 5 «Сторонние инструменты для wxWidgets»).

1.5 wxWidgets и объектно-ориентированное программирование

Как и большинство современных фреймворков, wxWidgets активно использует концепции объектно-ориентированного программирования. Каждое окно представляется в виде объекта C++. Эти объекты имеют определенное поведение, могут получать различные сообщения и реагировать на них. То, что в результате видит пользователь, является визуальным отображением взаимодействия системы объектов. Вы, как программист, должны управлять коллективным поведением объектов. Эту задачу выполнить достаточно легко, так как некоторое стандартное поведение элементов библиотека wxWidgets уже реализовала за вас.

Конечно, связав вместе объектно-ориентированный подход и GUI, можно достичь неплохих результатов. Объектно-ориентированный язык Smalltalk, разработанный в 1970-м году Аланом Каем (Alan Kay), стал важной вехой в развитии GUI, принес инновации в технологию пользовательских интерфейсов и структуры языка, и хотя wxWidgets использует другой язык и API, но основные принципы построения по большей части остались теми же.

1.6 Лицензионное соглашение

Лицензия wxWidgets (официально, по историческим и юридическим причинам называемая «лицензией wxWindows») является лицензией LGPL с некоторыми изменениями. Вы можете подробно ознакомиться с текстом лицензии, который доступен на сайте проекта или в каталоге docs дистрибутива, однако вкратце его суть такова: вы можете использовать wxWidgets в коммерческих или открытых проектах абсолютно бесплатно. Вы можете линковать библиотеку как статически, так и динамически. В любом из случаев, вы не обязаны делать доступными ваш собственный исходный код или объектные файлы, однако если вы делаете изменения в коде самой библиотеки, то вы обязаны предоставить всем доступ к этим изменениям бесплатно. Также желательно ознакомиться с лицензией отдельных компонентов, являющихся частью wxWidgets, таких как библиотеки по работе с PNG и JPEG.

Исходный код, который поставляется с данной книгой, публикуется под лицензией wxWindows.

Таблица 1.1: API портов wxWidgets

wxWidgets API								
Порт wxWidgets								
wxMSW	wxGTK	wxX11	wxMotif	wxMac	wxCocoa	wxOS2	wxPalmOS	wxMGL
API целевой платформы								
Win32	GTK+	Xlib	Motif/Lesstif	Carbon	Cocoa	PM	Palm OS Protein APIs	MGL
Операционная система								
Windows/ Windows CE	Unix/Linux			Mac OS 9/Mac OS X	Mac OS X	OS/2	Palm OS	Unix/ DOS

1.7 Архитектура wxWidgets

Таблица 1.1 показывает четыре концептуальных уровня: внешний API wxWidgets, основной порт, API платформы, используемый этим портом и, наконец, название операционной системы.

Далее мы перечислим основные порты wxWidgets, которые существовали на время написания книги.

1.7.1 wxMSW

Этот порт компилируется и запускается на всех 32-х и 64-х битных версиях Microsoft Windows, включая Windows 95, Windows 98, Windows ME, Windows NT, Windows 2000, Windows XP и Windows 2003. Он также может быть скомпилирован при использовании библиотеки Winelib под Linux. Также существует рабочая конфигурация под Windows CE (см. «wxWinCE»). Порт wxMSW может быть сконфигурирован для использования wxUniversal, вместо стандартных Win32 элементов.

1.7.2 wxGTK

wxWidgets для GTK+ может использовать элементы GTK+ версии 1.x или 2.x на любой из разновидностей Unix, поддерживающих X11 и GTK+ (например, Linux, Solaris, HP-UX, IRIX, FreeBSD, OpenBSD, AIX и других). Библиотека также может работать на встраиваемых платформах с достаточными для этого ресурсами, например, в среде GPE Palmtop (смотрите рис. 1.4). wxGTK является рекомендуемым портом для всех Unix-систем.

1.7.3 wxX11

wxWidgets для X11 использует набор элементов из wxUniversal и запускается непосредственно в Xlib, которая не имеет собственных элементов управления. Это обстоятельство делает порт подходящим для встраиваемых систем, но при его можно использовать и для построения приложений, которым не хотелось бы иметь зависимости от GTK+. Библиотека позволяет запускать приложение на любой Unix-системе,

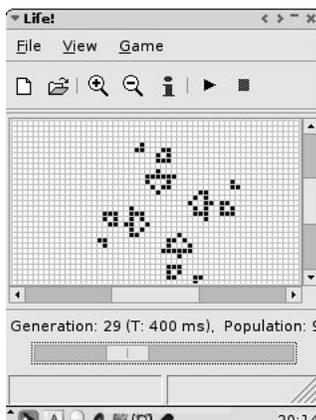


Рис. 1.4: Игра «Life!» из поставки wxWidgets под GPE на iPAQ PDA

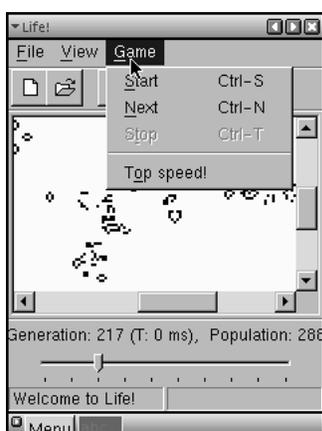


Рис. 1.5: Игра «Life!» из поставки wxWidgets, скомпилированная с помощью wxX11

где может быть запущен X11. Порт wxX11 не является таким же развитым, как и wxGTK. На рис. 1.5 показана игра «Life!», скомпилированная вместе с портом wxX11 и запущенная в системе Familiar Linux/TinyX на iPAQ PDA.

1.7.4 wxMotif

Этот порт может использовать Motif, OpenMotif или Lesstif на большинстве Unix-систем. Sun Microsystems оценивает функциональность своей библиотеки между GTK+ и GNOME, поэтому использование Motif не является желательной опцией для большинства разработчиков и пользователей.

1.7.5 wxMac

wxMac нацелен на запуск приложений в среде Mac OS 9 (требуется версия не меньшая 9.1) и Mac OS X (требуется версия не меньшая 10.2.8). Чтобы компилировать приложения для Mac OS 9 вам потребуются утилиты из Metrowerks CodeWarrior. Для Mac OS X вы можете использовать как Metrowerks CodeWarrior, так и Apple



Рис. 1.6: Игра «Life!» на Pocket PC 2003

tools. В последнем случае вам необходимо использовать Xcode версии 1.5 или выше. Если вы любите пользоваться командной строкой, то можно обойтись компилятором GCC версии 3.3 или выше.

1.7.6 wxCocoa

Этот порт еще находится в процессе разработки и предназначен для использования Cocoa API, впервые появившейся в Mac OS X. Хотя функциональность Carbon и Cocoa очень похожа, но этот порт имеет потенциальную возможность поддержки среды GNUStep на платформах, отличных от Mac.

1.7.7 wxWinCE

Порт Windows CE является оболочкой над различными SDK для систем на базе Windows CE, включая Pocket PC and Smartphone. Основной частью данного порта является порт для Win32 wxMSW с ограничениями и добавлениями для работы на малых платформах. Рис. 1.6 показывает игру «Life!», запущенную в эмуляторе Pocket PC 2003. На рис. 1.7 показаны четыре экрана примера «dialog» из поставки wxWidgets, запущенного на Smartphone 2003 с экраном 176x220 пикселей. Как вы видите wxWidgets сделал адаптацию приложения для этих платформ, включающую создание специального меню, вместо обычного, так как Smartphone поддерживает только две кнопки в меню. Требуются дополнительные подсказки от приложения, такие как вызов SetLeftMenu и SetRightMenu вместо добавления кнопок OK и Cancel в диалог.

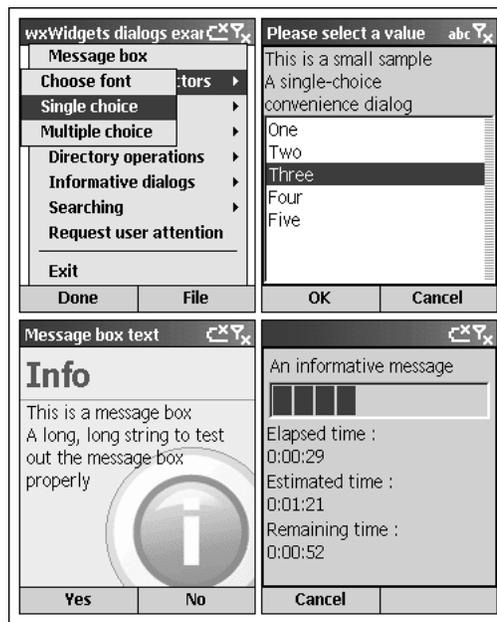


Рис. 1.7: Пример «dialogs» из поставки wxWidgets на Smartphone 2003

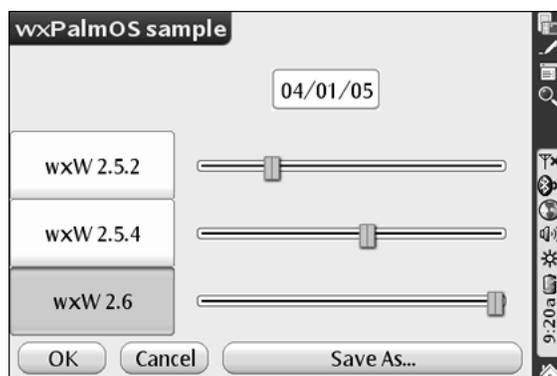


Рис. 1.8: Простейшее приложение под Palm OS 6

1.7.8 wxPalmOS

Этот порт предназначен для Palm OS 6 (Cobalt). В данный момент (на момент написания книги) он практически нефункционален, однако может быть использован для компиляции и запуска примера «simple» в эмуляторе Palm OS 6 (смотрите рис. 1.8).

1.7.9 wxOS2

Порт wxOS2 является менеджером отображения для OS/2 или eComStation.

1.7.10 wxMGL

Этот порт направлен на использование низкоуровневого графического слоя MGL от SciTech Software, Inc. и использует элементы wxUniversal.

1.8 Внутренняя организация

Внутри код wxWidgets разделен на шесть слоев:

- Общий код используется всеми портами. Он включает в себя классы структур данных, информацию о типах времени выполнения и описание базовых классов, таких как wxWindowBase, которые используются для сокращения кода во всех реализациях классов.
- Производный код реализует расширенные элементы управления независимо от платформы, позволяя эмулировать эти элементы и их функциональность, если они не присутствуют на данной платформе. wxWizard и wxCalendarCtrl являются примером таких элементов.
- wxUniversal — это множество базовых элементов для платформ, у которых эти элементы отсутствуют, таких как X11 и MGL.
- Код, зависящий от платформы, реализует классы, используя существующую функциональность. Например, wxMSW реализует wxTextCtrl как оболочку над обычным элементом редактирования для Win32.
- Дополнительный код находится в отдельном каталоге с именем contrib и включает в себя не обязательные, но полезные классы, такие как wxStyledTextCtrl.
- Сторонний код состоит из библиотек, разрабатываемых независимо от библиотеки wxWidgets, но которые используются ею для реализации важных возможностей. Сторонний код включает в себя библиотеки JPEG, Zlib, PNG и Expat, а также некоторые другие.

Каждый порт берет все, что ему необходимо из этих уровней и реализует API библиотеки wxWidgets.

Как wxWidgets узнает какие классы необходимо использовать во время компиляции? Когда вы включаете в программу заголовочные файлы wxWidgets, такие как wx/textctrl.h, в программу включается определяемый целевой платформой файл (в случае с Windows это файл wx/msw/textctrl.h) в соответствии с директивами в файле wx/textctrl.h. Далее ваша программа линкуется вместе с библиотекой, которая была скомпилирована с подходящими настройками. Одновременно может существовать несколько конфигураций библиотеки, в частности, версии Debug и Release, а также динамическая и статическая версии библиотеки. У вас есть возможность выключить некоторые компоненты при сборке wxWidgets, или сделать выбор между сборками с поддержкой Unicode или ANSI, редактируя для этого файл setup.h или используя опции для конфигурации, вид которых зависит от используемого вами компилятора. Дополнительная информация доступна в Приложении 1 «Установка wxWidgets».

Обращаем ваше внимание на то, что хотя wxWidgets и является оболочкой над каждым родным API, но она не запрещает писать специфический для платформы код, если это необходимо. Однако такая необходимость возникает очень редко.

1.9 Итоги

В этой главе вы узнали, что такое библиотека wxWidgets, историю ее развития, назначение всех доступных портов и немного о внутреннем устройстве библиотеки. В следующей главе мы подробно разберем наше первое приложение, написанное с использованием wxWidgets.