

Руководство пользователя Wine

Оригинал: [Wine User Guide](#)

Перевод: Алексей Дмитриев, yakwiat@yandex.ru

Дата перевода: май 2008

Содержание

[Глава I. Введение](#)

[1.1. Обзор \(про что этот документ\)](#)

[1.1.1. Цель данного документа и предполагаемая аудитория его читателей](#)

[1.1.2. Дальнейшие вопросы и комментарии](#)

[1.1.3. Краткий обзор. Основные шаги.](#)

[1.1.4. Быстрый старт](#)

[1.2. Что такое Wine](#)

[1.2.1. Windows и Линукс](#)

[1.2.2. Что такое Wine, и чем он мне поможет?](#)

[1.2.3. Свойства Wine](#)

[1.3. Версии Wine](#)

[1.3.1. Wine от Wine HQ](#)

[1.3.2. Другие версии Wine](#)

[1.4. Другие способы достичь того же результата](#)

[1.4.1. Аналогичные оригинальные приложения](#)

[1.4.2. Другие Операционные Системы](#)

[1.4.3. Виртуальные машины](#)

[Глава II. Установка Wine](#)

[2.1.1. Установка из пакетов](#)

[2.1.2. Установка из архива исходного кода](#)

[2.1.3. Установка из Git tree](#)

[2.2. Установка Wine из пакета](#)

[2.2.1. Установка "свежего" пакета](#)

[2.2.2. Различные дистрибутивы](#)

[2.3. Установка из исходного кода](#)

[2.3.1. Установка зависимостей](#)

[2.3.2. Компиляция Wine](#)

[2.3.3. Удаление Wine, установленной из исходников](#)

[Глава III. Настройка Wine \(Конфигурация\)](#)

[3.1. Программа Winecfg](#)

[3.1.1. Настройки Приложений](#)

[3.1.2. Настройка Библиотек](#)

[3.1.2.1. Замена DLL](#)

[3.1.2.2. По поводу системных DLL](#)

[3.1.2.3. Отсутствующие DLL](#)

[3.1.3. Графические настройки \(Graphics Settings\)](#)

[3.1.4. Настройки дисков](#)

[3.1.5. Аудио настройки](#)

[3.1.6. Компоновка рабочего стола](#)

[3.2. Использование Регистра и программы Regedit](#)

[3.2.1. Структура регистра](#)

[3.2.2. Файлы регистра Wine](#)

[3.2.3. Как пользоваться Regedit](#)

[3.2.4. Подсказки для системного администратора](#)

[3.2.5. Полный список ключей регистра](#)

[3.3. Прочие аспекты, поддающиеся настройке](#)

[3.3.1. Последовательные и параллельные порты](#)

[3.3.2. Совместные сети](#)

[3.3.3. Шрифты](#)

[3.3.4. Принтеры](#)

[3.3.5. Сканеры](#)

[3.3.6. Базы данных ODBC](#)

[3.3.6.1. Настройка ODBC в Unix](#)

[3.3.6.2. Использование ODBC драйверов Windows](#)

[Глава IV. Работа с Wine](#)

[4.1. Основное использование: приложения и апплеты контрольной панели](#)

[4.2. Как запускать Wine](#)

[4.3. Графический интерфейс Wine, наподобии "Проводника" Windows](#)

[4.4. Опции командной строки Wine](#)

[4.4.1. --help](#)

[4.4.2. --version](#)

[4.5. Переменные окружения](#)

[4.5.1. WINEDEBUG=\[channels\]](#)

[4.5.2. WINEDLLOVERRIDES=\[DLL Overrides\]](#)

[4.5.3. Настройка OSS звуковых драйверов](#)

[4.6. Опции командной строки wineserver'a](#)

[4.6.1. -d<n>](#)

[4.6.2. -h](#)

[4.6.3. -k\[n\]](#)

[4.6.4. -p\[n\]](#)

[4.6.5. -w](#)

[4.7. Настройка переменных окружения Windows/DOS](#)

[4.8. Программы с текстовым выводом \(CUI: консольный интерфейс пользователя\)](#)

[4.8.1. Настройка CUI исполняемых файлов](#)

[Глава V. Наладка и сообщение о найденных ошибках](#)

[5.1. Что делать, если программа не работает?](#)

[5.1.1. Проверьте конфигурацию своей Wine](#)

[5.1.2. Попробуйте разные версии Windows](#)

[5.1.3. Используйте различные пути \(paths\) для запуска](#)

[5.1.4. Поэкспериментируйте с настройкой DLL](#)

[5.1.5. Проверьте рабочее окружение!](#)

[5.1.6. Используйте другой режим графического интерфейса](#)

[5.1.7. Проверьте приложение!](#)

[5.1.8. Измените окружение Wine](#)

[5.1.9. Переконфигурируйте Wine](#)

[5.1.10. Следите за сторонней информацией](#)

[5.1.11. Ищите причину!](#)

[5.2. Как послать отчет об ошибке](#)

[5.2.1. Отчеты об ошибках](#)

[5.2.2. Аварийное завершение программы](#)

[5.2.2.1. Простой путь](#)

[5.2.2.2. Трудный путь](#)

[Словарь](#)

[Таблица 1. Отладочные каналы](#)

[Таблица 2. Основные различия в консолях](#)

[Таблица 3. Опции настройки Wineconsole](#)

Глава I. Введение

1.1. Обзор (про что этот документ)

1.1.1. Цель данного документа и предполагаемая аудитория его читателей

Этот документ, озаглавленный "Руководство пользователя Wine", является одновременно простым руководством по установке и расширенным справочником. Этот документ предназначен как для новичка, так и для опытного пользователя программы Wine, содержит как полные пошаговые инструкции по установке и конфигурации, так и подробные справочные материалы, документируя все настройки и средства поддержки.

1.1.2. Дальнейшие вопросы и комментарии

Если, после изучения этого руководства, списка FAQ, и других прилагающихся документов, осталось еще что-либо непонятное, мы с радостью выслушаем вас. [Раздел подписных листов](#) содержит несколько листов рассылки и IRC канал, каждый из которых - прекрасный способ найти помощь и сделать заявления. Если же вы являетесь компетентным человеком и уверены, что нечто может быть объяснено вами лучше, вы можете послать сообщение об ошибках <http://bugs.winehq.org/> или [послать patch](#) (заплатку) к документации Wine непосредственно нам.

1.1.3. Краткий обзор. Основные шаги.

Чтобы иметь возможность использовать Wine, у вас должна иметься работающая операционная система. Это руководство поможет превратить вашу систему из пустой, без Wine'ной области в область, гордящуюся свежей, самой современной установкой Wine.

- Первый шаг: [Установка Wine](#) описывает разнообразные методы установки файлов Wine в ваш компьютер.
- Следующий шаг: [Настройка Wine](#) расскажет, как настроить Wine под ваши индивидуальные нужды.
- Финальный шаг: [Использование Wine](#), описывает способы, как заставить определенное приложение работать под Wine лучше, а также предоставит полезные ссылки, на случай, если вам нужна помощь.

1.1.4. Быстрый старт

Процесс установки и запуска Wine может быть суммирован следующим образом:

- Загляните на страницу [Wine Downloads](#) и установите программу как указано в главе [Установка Wine](#). Для случайного или начинающего пользователя предпочтительней выбрать rpm пакет.
- Настройте Wine, используя команду `winecfg`. В большинстве случаев Wine теперь будет пригоден к работе.
- Чтобы проверить вашу установку, запустите файловый менеджер Wine, похожий на файловый менеджер Windows 3.1, используя команду `wine winefile`.
- Запустите Wine при помощи команды `wine filespec/appname.exe`
- Первая команда, которую вы запустите, будет команда установки какой-либо программы. Обычно что-то вроде: `wine /media/cdrom/setup.exe`, или нечто подобное.

1.2. Что такое Wine

1.2.1 Windows и Линукс

Различные программы созданы для различных операционных систем, и большинство из них не будут работать на системах, для которых они не предназначены. Программы под Windows, например не станут работать под Линукс, так как содержат инструкции, "не понятные" системе, пока они не "переведены" средой Windows. Точно также, Линукс программы не запустятся под ОС Windows, потому что Windows неспособна понять и интерпретировать все их инструкции.

Для того, кто желает пользоваться Windows и Линукс программами вместе, такая ситуация представляет собой фундаментальную проблему. Обычно проблема решается установкой и Windows и Линукс на один и тот же компьютер, что получило название "двойная загрузка". Когда нужна Windows программа, пользователь загружает компьютер в Windows и запускает программу; Когда нужна Линукс программа, пользователь перезагружает компьютер в Линукс. Эта практика весьма неудобна - не только пользователь вынужден терпеть неудобство частой перезагрузки машины, но и программы с разных платформ не могут работать одновременно. Кроме того, наличие Windows в числе операционных систем на компьютере создает дополнительные трудности: программное обеспечение дорого, требуется отдельный раздел жесткого диска, Windows "не видит" большинство форматов файловых систем, усложняя обобществление информации между операционными системами.

1.2.2. Что такое Wine, и чем он мне поможет?

Wine позволяет запускать Windows программы на любых Юникс-подобных операционных системах, особенно Линукс. В своей сути, Wine представляет собой реализацию библиотеки интерфейса программирования приложений (англ. Application Programming Interface, API) Windows, работающий подобно мосту между Windows программой и Линукс. Считайте Wine прослойкой совместимости; когда Windows программа пытается выполнить некое действие, которое в норме непонятно для Линукс, Wine "переведет" инструкцию этой программы на язык, понятный для системы Линукс. Например, если программа просит систему создать пусковую кнопку, или текстовое поле, то Wine преобразует Windows инструкции в их Линукс эквивалент в форме команды оконному менеджеру, используя стандартный протокол X11.

Если имеется доступ к исходному коду Windows программы, Wine также может быть использована для перекомпиляции программы в формат, который Линукс понимает намного проще. Для запуска такой перекомпилированной программы, Wine все равно необходим, однако у компиляции Windows программ прямо внутри Линукс существует масса преимуществ. Смотрите Winelib User Guide для знакомства с этой темой.

1.2.3. Свойства Wine

В процессе своего развития Wine постоянно приобретает новые свойства и увеличивает число программ, которое может запускать. Вот частичный список этих свойств:

- Поддержка запуска Win32 (Win 95/98, NT/2000/XP), Win16 (Win 3.1) и DOS программ
- Факультативная возможность использования DLL файлов от внешнего разработчика (таких, какие входят в состав Windows)
- Графический дисплей, базирующийся на X11, позволяющий удаленный доступ на любой X терминал, как и на текстовую консоль
- Рабочий стол внутри окна, либо смешиваемые окна
- Поддержка DirectX для игр
- Неплохая поддержка для различных звуковых драйверов, включая OSS и ALSA
- Поддержка альтернативных устройств ввода

- Печать: драйверы интерфейса PostScript (psdrv) для стандартного печатного сервиса PostScript для Юникс
- Модем, поддержка последовательного порта
- Поддержка сетевого протокола Winsock TCP/IP
- Поддержка для сканеров, пишущих CD приводов, и других устройств, использующих интерфейс ASPI (SCSI)
- Продвинутая поддержка Юникода и иностранных языков
- Полнофункциональная система отладки, настройки, фиксирования сбойных сообщений о неполадках для простоты поиска неисправностей

1.3. Версии Wine

1.3.1. Wine от Wine HQ

Примечание переводчика: как я полагаю под "HQ" имеется в виду Headquarters (Штаб-квартира) .

Wine является проектом с открытыми исходными кодами, и у вас есть чрезвычайно широкий выбор различных версий. Стандартная версия Wine появляется в виде периодических релизов (примерно дважды в месяц), и может быть скачана в Интернете, как в виде пакетированного двоичного файла, так и в виде исходного кода, готового к компиляции. Кроме того, вы можете установить версию, находящуюся в разработке, используя новейший из исходных кодов в репозитории Git. Загляните в следующую главу, [Getting Wine](#), для подробного описания.

1.3.2. Другие версии Wine

Существуют множество программ, так или иначе ведущих свое происхождение от стандартной кодовой базы Wine. Некоторые из них являются коммерческими продуктами, произведенными компаниями, активно поддерживающими проект Wine.

Эти продукты пытаются выделяться и отличаться от стандартной версии Wine, предлагая лучшую совместимость, упрощенную конфигурацию, и коммерческую поддержку. Если вам требуются подобные вещи, может быть вам стоит приобрести подобные продукты.

Вот пара предложений такого рода:

[CodeWeavers CrossOver Office](#)

CrossOver Office позволяет вам установить ваши любимые Windows приложения в среду Линукс, для чего не требуется лицензия Микрософт (Microsoft Operating System license). CrossOver имеет простой в использовании интерфейс, позволяющий устанавливать Windows приложения легко и быстро. Способ распространения: коммерческий; тридцатидневный период полнофункциональной демонстрации.

[CodeWeavers CrossOver Office Server Edition](#)

CrossOver Office Server Edition позволяет вам запускать ваши любимые Windows приложения в распределенном клиент-терминальном рабочем окружении под Линуксом, без необходимости приобретать лицензию Микрософт (Microsoft Operating System licenses) для каждой клиентской машины. CrossOver Office Server Edition позволяет вам удовлетворить запросы буквально сотен параллельных пользователей, и все это с одного сервера.

1.4. Другие способы достичь того же результата

Существует множество способов запускать программы, кроме Wine. Если вы планировали использовать Wine для запуска какой-либо программы, но нечто помешало вам, стоит обдумать и другие подходы к проблеме.

1.4.1. Аналогичные оригинальные приложения

Вместо использования конкретного Windows приложения при помощи Wine, можно просто взять другую программу. Множество Windows приложений, таких как медиа-плееры, средства мгновенного обмена сообщениями, программы совместного использования файлов, имеют очень хорошие свободные эквиваленты. Больше того, значительное число Windows программ было прямо портировано в Линукс, что делает ненужным ни Wine, ни самое Windows.

1.4.2. Другие Операционные Системы

Наиболее очевидный способ запустить Windows приложение - это запустить его под Windows. Однако соображения безопасности, стоимости лицензий, обратной совместимости, эффективности использования машин, могут осложнить это предложение. Именно поэтому так полезен Wine.

Можно попробовать использовать [ReactOS](#), которая является полной свободной (с открытыми кодами) альтернативой Windows. ReactOS тесно сотрудничает (и обменивается кодами) с проектом Wine, но запускает Windows приложения на базе ядра ReactOS, а не на базе Linux. Кроме того, ReactOS предлагает совместимость с файлами драйверов Windows, что позволяет использовать "железо" без функциональных Линукс драйверов.

1.4.3. Виртуальные машины

Вместо того чтобы устанавливать новую операционную систему на свою машину, можно, вместо этого, запустить виртуальную машину на программном уровне, и установить новую операционную систему на нее. Таким образом, вы сможете пользоваться системой Линукс, и в то же время, в виртуальной машине, пользоваться Windows со всеми его программами, одновременно и на одном и том же "железе". Виртуальные машины позволяют вам устанавливать и запускать не только различные версии Windows, на одном и том же оборудовании, но также другие операционные системы, включая ReactOS.

Предлагаются несколько различных типов виртуальных машин, некоторые даже могут эмулировать "железо" x86 на иных платформах. Из машин со свободными исходниками упомяну [Bochs](#) и [QEMU](#), способных виртуально запускать Windows и ReactOS. Другие, коммерческие виртуальные машины включают [VMware](#) и Микрософтовский [VirtualPC](#).

В работе с виртуальными машинами есть существенные недостатки. В отличие от Wine, эти программы являются эмуляторами, так что при работе с ними неизбежна потеря скорости, которая может быть весьма ощутима. Более того, запуск приложения внутри виртуальной машины, не позволяет полностью интегрировать приложение в рабочее окружение. Например, вы хотите иметь значки системного трея Windows, или ярлыки приложений Windows среди значков Линукс, а на деле получается, что элементы Windows могут существовать только внутри виртуальной машины.

Глава II. Установка Wine

2.1.1. Установка из пакетов

Наиболее простой метод установки Wine - это использовать подготовленный пакет. Такие пакеты содержат готовые к работе двоичные файлы Wine, скомпилированные специально для вашего дистрибутива, они регулярно проверяются соответствующими службами на работоспособность и самодостаточность (полноту).

Пакетный метод установки рекомендуется для установки Wine. Мы делаем их легко доступными на страничке загрузки [WineHQ downloads page](#), и это всегда самые новые из доступных пакетов. Будучи весьма популярными, пакеты Wine часто встречаются в репозиториях различных дистрибутивов. Данные пакеты могут быть несовременными, это зависит от дистрибутива. Но пакеты легко обновить, во многих дистрибутивах это делается несколькими кликами мыши. Создание своего собственного, пригодного для инсталляции, бинарного пакета из исходного кода также возможно, хотя и выходит за рамки данного руководства.

2.1.2. Установка из архива исходного кода

Случается, что пакет Wine не вполне устраивает вас. Возможно, он не подходит вам по архитектуре или дистрибутиву, или, возможно вы хотите построить пакет, используя собственные настройки компилятора, или отключить некоторые опции, а может быть вам необходимо модифицировать часть исходного кода, прежде чем компилировать его. Так как вы работаете со свободными исходниками, вы вольны проделывать все эти манипуляции с исходным кодом, который регулярно поставляется с каждым новым релизом Wine. Этот способ установки подразумевает скачивание архива исходного кода Wine, и компиляцию из командной строки. Если вы умеете это делать и имеете в этом нужду, такая возможность, видимо, для вас.

Получить архив исходных кодов очень просто. Каждый релиз снабжается пакетом исходников в формате сжатия tar.bz2 на странице [WineHQ downloads page](#). Компиляция и установка Wine из исходников несколько сложнее установки из пакетов, однако мы поможем преодолеть трудные места и поддержать вас на этом пути.

2.1.3. Установка из Git tree

Если вы хотите оказаться на переднем крае разработки Wine, или даже принять в разработке участие, вы можете закачать самый последний исходный код из репозитория Git. Инструкции как это сделать вы найдете на сайте [Git](#).

Пожалуйста, имейте в виду, что все обычные предупреждения о разрабатываемых версиях остаются в силе. Исходники в репозитории Git обычно не проверены, и могут даже не компилироваться правильно. Тем не менее, это лучший способ проверить, как будет работать Wine в следующей версии, а если вы модернизируете исходный код, то лучше это делать на новейших образцах. Репозиторий Git также полезен распространителям приложений, заинтересованных в том, чтобы приложение работало и со следующим релизом, или что последний патч по-прежнему делает свое дело. Если вы хотите помочь нам сделать приложения работоспособными в Wine, загляните в [HowTo](#).

2.2 Установка Wine из пакета

2.2.1. Установка "свежего" пакета

Установка "свежего" пакета на свежую систему невероятно проста. Просто скачайте и установите пакет, используя предлагаемую вашим дистрибутивом утилиту. Даже обычно не требуется удалять старый пакет, так как современные дистрибутивы Линукс должны обновлять и заменять их автоматически. Если вы прежде устанавливали Wine из исходного кода, то вам следует удалить его до установки пакета. Смотрите раздел [деинсталляция Wine из исходного кода](#) для получения правильных инструкций.

2.2.2. Различные дистрибутивы

Wine работает на большом числе различных дистрибутивов Линукс, как и на прочих Юникс-подобных системах, таких, как Solaris и FreeBSD, каждая со своими специфическими способами установки и управления пакетами. К счастью, основные принципы у них всех схожи, и установка Wine будет не сложнее, чем установка прочего программного обеспечения, независимо от дистрибутива.

Деинсталляция пакетов Wine столь же проста, и обычно происходит через те же несложные интерфейсы, что и установка.

Мы не станем вдаваться в подробности установки и удаления пакетов Wine в различных дистрибутивах, вы всегда можете найти современные инструкции для конкретных дистрибутивов на сайте WineHQ в разделе HowTo. Если вам нужна дальнейшая помощь, учитывая, насколько легко устанавливаются пакеты Wine, то мы советуем обратиться к документации, на форумы поддержки и каналы IRC.

2.3 Установка из исходного кода

Прежде, чем начать установку из исходного кода, убедитесь, что деинсталировали все до одного бинарные пакеты, которые могли быть в вашей системе. Установка из исходников подразумевает использование терминала и полной копии исходного кода Wine. Скачав исходники с Git, либо распаковав их из архива, переходите в их директорию при помощи терминала и приступайте к следующим шагам.

2.3.1 Установка зависимостей

Wine использует множество библиотек в процессе своей работы. В то же время, Wine жестко не зависит от них, и может быть скомпилирована без большинства из них; но функциональность Wine обеспечивается наличием этих библиотек в процессе компиляции. Раньше многие проблемы пользователей были обусловлены отсутствием необходимых библиотек при сборке Wine из исходного кода. По этой и ряду других причин, мы настойчиво рекомендуем устанавливать Wine их бинарных пакетов, либо создавать пакеты исходников, способные автоматически отслеживать зависимости при сборке.

Если вы хотите установить зависимости вручную, у вас есть несколько способов заметить отсутствие необходимых библиотек. Самый прямой путь - просматривать вывод программы `configure`, прежде начала компиляции, и смотреть, чего важного не хватает. Затем просто установить то, что не хватает, и снова запустить программу `configure`, прежде чем начинать компилировать. Можно также исследовать файл, который генерирует программа `configure` (`include/config.h`) и выяснить, какие файлы безрезультатно ищет программа `configure` (и не находит).

2.3.2. Компиляция Wine

Закончив с установкой зависимостей, вы готовы начать компиляцию. В окне терминала, перейдя в директорию с деревом исходного кода программы, запустите следующие программы:

```
$ ./configure
$ make depend
$ make
# make install
```

Последняя команда требует привилегий суперпользователя. Хотя вы никогда не будете запускать Wine как `root`, для установки это совершенно необходимо.

2.3.3. Удаление Wine, установленной из исходников

Для деинсталляции Wine, установленной из исходников, снова переместитесь в директорию, в которой находятся исходники, и которой вы пользовались для установки с помощью терминала. Затем запустите команду:

```
# make uninstall
```

Эта команда потребует привилегий суперпользователя, и должна удалить все двоичные файлы Wine из вашей системы. Однако она не тронет файлы конфигурации Wine, а также приложения, расположенные в вашей домашней директории, так что вам решать - удалять файлы вручную, либо устанавливать следующую версию поверх.

Глава 3. Настройка Wine (Конфигурация)

Подавляющее большинство обычных настроек делается через инструмент Winecfg tool. Мы пройдем простое пошаговое введение в Winecfg и рассмотрим доступные опции. В следующем разделе мы займемся более продвинутыми изменениями, которые можно сделать при помощи regedit. Также мы проведем полный обзор всех конфигурационных настроек Wine. И, под конец, мы рассмотрим настройки (которые вы, возможно, захотите сделать) недоступные через Winecfg и regedit.

3.1. Программа Winecfg

В прошлом, Wine использовал специальный конфигурационный файл, находившийся в ~/.wine/config. Если вы до сих пор пользуетесь версией Wine, имеющей дело с этим файлом (старше июня 2005 года), то вам нужно обновиться, прежде чем делать что-либо дальше. Все настройки теперь собраны прямо в registry (реестре) и доступны Wine при запуске.

Программа Winecfg должна быть установлена на ваш компьютер вместе с остальными программами Wine. Если вы не поймете, как запустить ее, попробуйте команду:

```
$ /usr/local/bin/winecfg
```

или просто

```
$ winecfg
```

Когда программа стартует, вы увидите сверху окна следующие вкладки:

- Applications
- Libraries
- Graphics
- Desktop Integration
- Drives
- Audio
- About

Внесение изменений во вкладки Applications и Libraries будет иметь решающее значение для возможности запуска приложений. Остальные настройки служат для того, чтобы заставить саму программу Wine работать так, как бы вам хотелось.

Предупреждение: Вкладки Applications, Libraries и Graphics связаны между собой! Если вы во вкладке Applications выберете Default Settings (настройки по умолчанию), все изменения, сделанные во вкладках Libraries и Graphics, будут изменены для всех приложений. Если вы настроили некое приложение во вкладке Applications и выбрали его, то настройки, произведенные во вкладках Libraries или Graphics, будут касаться только данного выбранного приложения. Это позволяет выборочно настраивать определенные приложения.

3.1.1. Настройки Приложений

Wine обладает способностью подражать поведению различных версий Windows. В целом, наиболее различны поведения версий Win9x и версий NT. Некоторые приложения требуют для нормальной работы специфического поведения, и изменения этой настройки может заставить сбойное приложение заработать. Недавно версией по умолчанию для Wine была избрана Windows 2000. Известно, что многие приложения работают лучше, если вы выберете Windows 98.

Внутри вкладки вы заметите строку Default Settings (Установки по умолчанию). Если вы выберете эту строку, то все приложения будут работать в версии Windows по умолчанию. Проблемное приложение лучше настраивать отдельно от установок по умолчанию. Для этого:

1. Нажмите кнопку Add application (добавить приложение).

2. Найдите нужный .exe файл.
3. После того, как вы добавите это приложение, вы сможете выбирать версию Windows, которую Wine будет эмулировать специально для этого приложения.

3.1.2. Настройка Библиотек

Подобным же образом, некоторые приложения требуют специальных библиотек. Wine воспроизводит систему библиотек Windows (так называемых "родных" DLL), создавая версии библиотек, которым не нужна лицензия Микрософт, и которые работают в точности как "родные". Wine испытывает недостаток в таких воссозданных версиях библиотек, но работают они вполне удовлетворительно. Использование только воссозданных DLL гарантирует независимость вашей системы от Микрософт. Однако Wine имеет возможность загружать и "родные" DLL Windows.

3.1.2.1. Замена DLL

Не всегда возможно запустить приложение, используя встроенные в Wine DLL. Иногда "родные" DLL просто лучше работают. После того, как вы "засекли" "родную" DLL в Windows системе, вы должны поместить ее в такое место, где Wine найдет ее, а затем настроить для работы. Обычно таковым местом является директория, которую вы назначили на роль `c:\windows\system32`. Существуют 4 DLL, "родные" версии которых никогда не следует использовать: `kernel32.dll`, `gdi32.dll`, `user32.dll` и `ntdll.dll`. Эти библиотеки требуют низкоуровневый доступ к ядру Windows, которого просто нет в структуре Wine.

Имея это в виду, коли вы уже скопировали DLL, то необходимо сообщить Wine о возможности воспользоваться им. Можно настроить Wine на выбор между "родной" и встроенной DLL на двух различных уровнях. Если у вас во вкладке Applications установлены Default Settings, то изменения, которые вы сделаете, коснутся всех приложений. Либо можно изменить глобальные установки на уровне отдельного приложения, добавив это приложение и выбрав его во вкладке Applications.

Чтобы добавить заменяющий файл `FOO.DLL`, впишите "FOO" в поле, озаглавленное *New override for library* (Новая замена для библиотеки), и нажмите кнопку *Add* (Добавить). Чтобы изменить поведение DLL, нужно выбрать ее из поля *Existing overrides*: (Существующие замены) и нажать *Edit*. По умолчанию, первой будет загружена "родная" Виндовская, второй - встроенная в Wine (*Native then Built-in*). Можно также выбрать только "родную", только встроенную, или отключить обе.

3.1.2.2. По поводу системных DLL

Команда Wine решила, что необходимо создать поддельные DLL файлы, чтобы обмануть многие программы, проверяющие только наличие файла, чтобы убедиться, что определенное свойство (такое как `Winsock` и его `TCP/IP networking`) теоретически доступно. Если для вас встала эта проблема, создайте в сконфигурированной директории `c:\windows\system32` пустые файлы с соответствующими именами, чтобы программа думала, будто они на месте, а встроенные в Wine DLL будут загружены, когда программа действительно их запросит. (К сожалению, инструмент `wineinstall` не создает такие пустые файлы самостоятельно).

Иногда приложения пытаются выяснить версию ресурса непосредственно из физического файла (скажем, чтобы узнать версию `DirectX`). Файлы пустышки в этом случае не помогут. В таком случае необходимо установить файлы с полным кодом версии. Эта проблема уже решена для многих файлов. Если нет, то вы опять должны скопировать "родной" DLL файл, чтобы обмануть программу.

Естественно, что существуют DLL, которые Wine еще не реализовала должным образом (или вовсе). Если под рукой нет настоящей Windows, с которой можно скопировать DLL, всегда можно получить файл с сайтов, содержащих архивы Windows DLL, которые вы найдете при помощи поисковых машин. Пожалуйста, соблюдайте правила лицензирования для тех файлов DLL, которые вы выбираете; некоторые распространяются без ограничений, некоторые нет.

3.1.2.3 Отсутствующие DLL

В случае, если Wine жалуется на отсутствие DLL, нужно проверить, является ли этот файл "официальным" DLL, или принадлежит программе, которую вы пытаетесь запустить. Это можно сделать, проведя в Интернете поиск по имени файла. После того, как вы "засекли" DLL, надо еще убедиться, что Wine сможет его использовать. Поиск DLL обычно производится в следующем порядке:

1. Директория, откуда программа была запущена.
2. Текущая (рабочая) директория.
3. Системная папка Windows.
4. Папка Windows.
5. Директории переменной PATH.

Вкратце: либо поместите нужный DLL в каталог вашей программы (может быть неудобно), либо поместите его в системную папку Windows. По возможности избегайте "родных" DLL на базе NT, так как в Wine поддержка NT API во много слабее, чем поддержка Win9x API (порой приводящая к более плохой совместимости с NT DLL, чем даже с не-Windows схемами!).

3.1.3 Графические настройки (Graphics Settings)

Имеется пять различных базовых настроек, которые можно конфигурировать. Для большинства людей достаточно и установок по умолчанию.

Первые несколько настроек имеют отношение в основном к играм и являются самоочевидными. Возможно запретить курсору мыши покидать пределы окна программы DirectX (т.е. игры). По умолчанию эта опция включена. Существует много причин сделать это, не последней из которых является тот факт, что проще играть, когда курсор заключен в меньшей площади. Другой резон подключить эту опцию - лучший контроль над мышью. Wine искажает позицию мыши, чтобы имитировать работу Windows. Подобным образом, "desktop double buffering" производит сглаживающие обновления экрана, отчего игры выглядят лучше; опять-таки, по умолчанию опция включена. Это компромисс повышенному потреблению памяти.

Может оказаться полезным эмулировать виртуальный рабочий стол. В этом случае все программы будут запускаться в отдельном окне. Это может помочь выявить глючные игры, которые изменяют разрешение экрана (возможно неудачно). Заключение их в окно может привести к лучшему контролю над ними, возможно ценой меньшего удобства. Можно попробовать размеры 640x480 (по умолчанию) или 800x600.

И последнее, можно изменить некоторые установки Direct3D. По большей части эти установки определяются автоматически, но вы можете заставить их вести себя определенным образом. Некоторые игры пытаются проверять систему на предмет поддержки тех или иных свойств. Выключив это поведение, Wine не будет сообщать о способности формировать изображение определенным образом. Это может привести к повышению скорости игры ценой качества графики; а может быть, игра вовсе не будет работать.

3.1.4. Настройки дисков

Windows подразумевает довольно жесткую схему обозначения дисков, которую имитирует Wine. Большинство людей знакомы со стандартной схемой: "A:" - представляет флорпи диск, "C:" - представляет первичный системный диск, etc. Wine использует ту же концепцию и размечает диски согласно основной "родной" файловой системе.

Разметка дисков в Wine относительно проста. В Winecfg во вкладке Drives вы видите кнопки для добавления и удаления нужных дисков. Когда вы выбираете "добавить диск" создается новая строка и появляется схема обозначения дисков по умолчанию. Можно изменить адрес, на который указывает данный диск, сделав изменения в поле Path. Если вы сомневаетесь в правильности Пути, нажмите кнопку "Browse" чтобы найти правильный Путь. Удалить диск можно, выделив нужный, и

нажав "Remove".

Winecfg может автоматически определить диски, доступные на вашей системе. Рекомендуется попробовать автоматический способ, прежде чем пытаться разметить диски вручную. Простым нажатием на кнопку Autodetect вы заставляете Wine искать диски в вашей системе.

Очень интересно настраивать диски, не прибегая к Winecfg, это замечательно просто. Все настройки дисков находятся в директории ~/.wine/dosdevices. Каждый диск - это просто симлинк к месту, где он в действительности находится. Wine автоматически устанавливает два диска во время первого запуска:

```
$ ls -la ~/.wine/dosdevices/  
lrwxrwxrwx 1 wineuser wineuser 10 Jul 23 15:12 c: -> ../drive_c  
lrwxrwxrwx 1 wineuser wineuser 1 Jul 23 15:12 z: -> /
```

Чтобы добавить еще диск, например CD-ROM, просто создайте еще один линк, указывающий на него:

```
$ ln -s /mnt/cdrom ~/.wine/dosdevices/d:
```

Обратите внимание на то, что в ссылках используется принятая в DOS система именования дисков - буква со знаком двоеточия, например, "a:". Так что, если линк к вашему диску c: указывает на ~/.wine/drive_c, вы можете понимать ссылки на c:\windows\system32 как ~/.wine/drive_c/windows/system32.

3.1.5 Аудио настройки

Wine может работать с весьма немногими звуковыми subsystemами, которые вы можете выбрать во вкладке "Audio". Winecfg выявит все доступные драйверы, но выбрать тот, что будет использоваться, вы можете вручную. Старые дистрибутивы Линукс, использовавшие ядро 2.4, обычно применяли "OSS" драйвер. Ядра 2.6 переключились, в большинстве случаев, на "ALSA". Драйвер "aRts" был недавно выведен из работы, из-за общей нехватки поддержки subsystemы "aRts". Если вы используете GNOME, можете попробовать Esound. OSS и ALSA аудио драйвера наиболее проверены, так что рекомендуется остановить свой выбор на них, если возможно. Если вам нужны "Jack", "NAS" или "CoreAudio", то, скорее всего, вы сами знаете почему.

Настройки DirectSound в основном используются в играх. Вы можете выбирать степень ускорения "железа", но для большинства прекрасно подойдет "Full" (Полное).

3.1.6 Компоновка рабочего стола

Wine может загружать темы Windows, если они у вас есть. Хотя никакой необходимости, с точки зрения работы Wine или приложений, в этом нет, это позволяет вам создавать привычный облик программы. Wine поддерживает новые MSStyles темы. В отличие от старых Microsoft Plus! style тем, механизм uxtheme поддерживает специальные .msstyles файлы, которые могут изменять все настройки Windows. Это примерно то же самое, что большинство Линукс десктопов умеет делать уже много лет. Если желаете попробовать, то делайте так:

1. Загрузите тему Windows XP. Убедитесь в наличие .msstyles файла.
2. Создайте цепочку директорий в своем ложном Windowsовском диске:

```
$ mkdir -p ~/.wine/drive_c/windows/Resources/themes/имя_темы
```
3. Переместите файл .msstyles в директорию /name-of-your-theme
4. При помощи вкладки Desktop Integration в winecfg выберите новую тему.

3.2 Использование Реестра и программы Regedit

Все настройки, производимые с помощью Winecfg, за исключением настройки дисков, в конце концов сохраняются в Реестре. В ОС Windows, это центральное хранилище настроек, как отдельных

приложений, так и самой операционной системы. Wine подобным же образом использует Реестр, поэтому некоторые настройки, которых нет в Winecfg, могут быть изменены в Реестре. (Гораздо больше шансов за то, что вам придется "залезать" в Реестр для изменения настроек какой-нибудь программы, чем для изменения настроек самой Wine).

Сам факт того, что Wine использует Реестр для сохранения настроек, является дискуссионным. Некоторые утверждают, что это слишком похоже на Windows. Чтобы возразить на это, нужно принять во внимание несколько моментов. Во-первых, невозможно обойтись без Реестра просто потому, что приложения "рассчитывают" сохранить свои настройки именно там. Для того чтобы Wine мог хранить и иметь доступ к настройкам в отдельном конфигурационном файле, потребуется отдельный набор программ, в основе своей делающий то же, что Win32 API, что Wine и делает. И, наконец, в отличие от Windows, Реестр Wine написан простым текстом и может быть изменен в любом текстовом редакторе. И вот, хотя большинство нормальных системных администраторов (и разработчиков Wine), ругаются как бешеные на запутанный Реестр Windows, все же необходимо, чтобы Wine каким-то образом поддерживал его.

3.2.1 Структура Реестра

Хотя это нам не совсем по пути, давайте немного углубимся в структуру реестра и посмотрим, как он устроен. Реестр Windows - это замысловатая древовидная структура, и даже лучшие Windows программисты не всегда знают, как он распланирован, со своими различными "ульями" ("hives") и бесконечными ссылками между ними; полное описание не входит в возможности данного документа. Но вот основные ключи Реестра вы теперь должны знать:

HKEY_LOCAL_MACHINE

Этот фундаментальный корневой ключ (в win9x он хранится в скрытом файле system.dat) содержит все, имеющее отношение к данной инсталляции Windows. Часто сокращается до HKLM.

HKEY_USERS

Этот фундаментальный корневой ключ (в win9x он хранится в скрытом файле user.dat) содержит информацию о каждом пользователе в данной инсталляции.

HKEY_CLASSES_ROOT

Это линк к HKEY_LOCAL_MACHINE\Software\Classes. Там хранится информация, описывающая вещи вроде ассоциаций файлов, OLE (Object Linking and Embedding) - управление документами, и классы COM (технология COM - стандарт Микрософт).

HKEY_CURRENT_USER

Это линк к HKEY_USERS\your_username, то есть к вашей персональной конфигурации.

3.2.2. Файлы реестра Wine

Теперь вы, возможно, задумались, каким образом перевести все это в структуры Wine. Планировка реестра, описанная выше, "проживает" в трех различных файлах в директории ~/.wine:

system.reg

Этот файл содержит HKEY_LOCAL_MACHINE.

user.reg

Этот файл содержит HKEY_CURRENT_USER.

userdef.reg

Этот файл содержит HKEY_HKEY_USERS\.Default
(то есть установки пользователя по умолчанию).

Эти файлы автоматически создаются программой wineprefixcreate во время первого запуска Wine. Набор основных установок сохраняется в файле `c:\windows\inf\wine.inf` и обрабатывается программой `rundll32.exe`. Когда вы впервые запускаете Wine, файл `rundll32.exe` подвергается процессу заполнения данными первоначального реестра. Чтобы ознакомиться с деталями, просмотрите скрипт `wineprefixcreate` и увидите, как все это делается. После обновления Wine `wineprefixcreate` снова может быть использован для обновления строк реестра по умолчанию.

Как уже говорилось, можно редактировать все эти `.reg` файлы, при помощи любого текстового редактора. Только убедитесь, что Wine не запущен в этот момент, иначе все ваши изменения пропадут.

3.2.3. Как пользоваться Regedit

Проще всего получить доступ и изменить Реестр при помощи инструмента `regedit`. Так же как одноименная Windows программа, которую он замещает, `regedit` предназначен для просмотра реестра на системном уровне, содержащем все его ключи. Просто запустите `regedit`, и он откроется. Вы сразу заметите, что таинственные ключи, представленные в текстовом файле, организованы иерархическим образом.

Чтобы перемещаться по реестру, кликните на ключах в левой части панели, чтобы "погрузиться" на следующий уровень глубины. Чтобы удалить ключ, кликните на нем и в меню `Edit` выберите `"Delete"`. Чтобы добавить ключ или значение, определите место, где вы хотите поместить его, и выберите `"New"` из меню `Edit`. Чтобы изменить существующий ключ, выделите его в правой панели и выберите `"Modify"` из меню `Edit`. Другой способ проделать то же самое, это щелчок правой кнопкой мыши на ключе или значении.

Особый интерес для пользователей Wine представляют установки в `HKEY_CURRENT_USER\Software\Wine`. Большинство настроек, что вы сделали при помощи `winecfg`, записаны именно в этот участок реестра.

3.2.4. Подсказки для системного администратора

Используя вышеописанную файловую структуру, можно настроить систему так, чтобы системная инсталляция Wine (с приложениями) могла быть общей для всех пользователей, и с то же время позволять каждому пользователю иметь свою конфигурацию. Администратор может, установив Wine и все необходимые для пользователей Windows приложения, затем скопировать получившийся файл `system.reg` и другие в файлы глобального реестра (которые, как мы договорились, находятся в `/usr/local/etc`), при помощи команд:

```
cd ~root/.wine
cp system.reg /usr/local/etc/wine.systemreg
```

и, возможно, даже дать ссылку оттуда обратно в администраторский аккаунт, чтобы в дальнейшем было проще устанавливать приложения для всей системы:

```
ln -sf /usr/local/etc/wine.systemreg system.reg
```

Вам, возможно, придется проделать все то же самое с файлом `user.reg`, несмотря на то, что этот файл содержит настройки пользователей. Каждый пользователь должен иметь свою собственную копию этого файла вместе с правами на его изменение.

Необходимо обратить внимание на разметку дисков. Если вы обобществляете файл `system.reg`, то вы должны убедиться, что установки реестра совместимы с разметкой дисков в файлах `~/.wine/dosdevices` каждого пользователя. Возьмите за правило, что, чем более ваша разметка дисков совпадает с таковой по умолчанию, созданной программой `wineprefixcreate`, тем легче ваша задача. Вы можете и не суметь сделать общими все диски `"c:"`, в которые вы первоначально устанавливали приложения. Некоторые приложения требуют возможности записать определенные настройки на диск, особенно приложения для Windows 95/98/ME.

Имейте в виду: если вы устанавливали Wine из исходников как суперпользователь, скрипт `tools/wineinstall`, которым вы пользовались при установке, больше не работает.

И последнее слово предупреждения: следите, что вы делаете с аккаунтом администратора. Если вы скопируете (или дадите линк) реестр администратора в глобальный реестр, то любой пользователь сможет читать настройки администратора, что плохо, если там содержится важная информация о паролях, персональная информация и прочее. Используйте аккаунт администратора только для установки программ, а не для ежедневной работы, ее нужно делать под обычным аккаунтом.

3.2.5. Полный список ключей реестра

Самый обновленный список употребительных ключей и значений вы найдете на странице [developer's wiki](#).

3.3. Прочие аспекты, поддающиеся настройке

Этот раздел содержит все остальные вещи, которые могут быть настроены. Он также служит собранием подсказок и хитростей, чтобы выжать все возможное из использования Wine.

3.3.1. Последовательные и параллельные порты

Последовательные и параллельные порты настраиваются почти как диски - просто создается символическая ссылка в директории `~/.wine/dosdevices` с именем устройства. В Windows последовательные порты принято называть "com" с добавлением номера, например: com1, com2, etc. Аналогично параллельные порты называются "lpt" с добавлением номера, например: lpt1. Вам следует связывать их напрямую ссылками с соответствующими устройствами Unix, такими как `/dev/ttyS0` и `/dev/lp0`. Для примера, чтобы настроить один последовательный и один параллельный порт запустите следующие команды.

```
ln -s /dev/ttyS0 com1
ln -s /dev/lp0 lpt1
```

3.3.2. Совместные сети

Сетевые ресурсы Windows могут быть распланированы в директории unc/, так что все, что должно быть в `\\myserver\some\file`, будет находиться в `~/.wine/dosdevices/unc/myserver/some/file/`. Например, если вы используете Samba чтобы смонтировать `\\myserver\some` на `/mnt/smb/myserver/some`, то вам следует выполнить команду:

```
ln -s /mnt/smb/myserver/some unc/myserver/some
```

чтобы сделать их доступными в Wine (только не забудьте создать директорию unc, если таковой не существует).

3.3.3. Шрифты

Настройка шрифтов, когда-то серьезная проблема, теперь решается намного проще. Если у вас в Windows есть набор TrueType шрифтов, то просто нужно скопировать все .ttf файлы в `c:\windows\fonts`.

3.3.4. Принтеры

Wine может взаимодействовать напрямую с вашей печатающей системой CUPS (система печати Юникс) и находить принтеры, доступные системе. Настройка принтеров в Wine сводится к тому, чтобы убедиться, что CUPS работает. Тем не менее, Wine нуждается в команде lpr (от CUPS), когда нужно распечатать какой-либо документ.

Если вы не пользуетесь CUPS, то применяется старая BSD-Printing система печати:

Все принтеры из файла `/etc/printcap` автоматически устанавливаются в Wine.

Wine требует PPD-File для каждого из принтеров (generic.ppd входит в состав Wine)

Для распечатки документа запускается команда `lpr`.

3.3.5. Сканеры

В ОС Windows, сканеры используют TWAIN API для доступа к соответствующим устройствам. Встроенный в Wine TWAIN DLL просто пересылает эти запросы на Linux SANE библиотеки. Итак, чтобы воспользоваться сканером под Wine, сначала нужно убедиться, что можете получить к нему доступ через SANE. После этого, нужно убедиться, что у вас есть `xscanimage` пригодный к использованию. Сейчас `xscanimage` входит в `sane-frontends` пакет, но может оказаться не установленным на вашем дистрибутиве. Вообще, доступ к сканерам и сегодня изобилует проблемами. Если у вас случайно сканер заработает, пожалуйста, обновите этот раздел Руководства пользователя Wine, подробно описав детали использования SANE с Wine.

3.3.6. Базы данных ODBC

Система Open DataBase Connectivity interface (открытый интерфейс взаимодействия с базами данных в Windows) также как и система печати, спроектирована, чтобы переключиться на Юникс систему на высоком уровне. Вместо того чтобы проверять работают ли Windows коды под Wine, она использует подходящий Юникс ODBC провайдер, такой как `UnixODBC`. Таким образом, если вы настроили Wine использовать встроенный `odbc32.dll`, этот Wine DLL будет взаимодействовать с вашим Unix ODBC пакетом, позволяя ему работать. Если же вы настроили Wine на использование "родной" (Виндовской) `odbc32.dll`, он будет пытаться использовать "родные" же ODBC32 драйвера и так далее.

3.3.6.1. Настройка ODBC в Unix

Первым шагом на пути использования системы Unix ODBC с Wine, будет, конечно, настройка работы самой системы Unix ODBC. Это может потребовать установки исходников или RPM пакетов. Существует несколько Unix ODBC систем; та, к которой привык автор, называется `unixODBC` (с IBM DB2 драйвером). Также существует мост ODBC-ODBC, который можно использовать для получения доступа к базе данных Microsoft Access. Обычно такие системы включают инструмент, например `isql`, который позволит вам получить доступ к данным из командной строки, так, чтобы вы имели возможность убедиться, что система работает.

Второй шаг - переключить Unix ODBC библиотеку на встроенную в Wine `odbc32 DLL`. Встроенная `odbc32` (в настоящее время) ищет в переменной окружения `LIB_ODBC_DRIVER_MANAGER` имя ODBC библиотеки. Для примера, у автора в файле `.bashrc` есть такая строка:

```
export LIB_ODBC_DRIVER_MANAGER=/usr/lib/libodbc.so.1.0.0
```

Если таковая переменная окружения не установлена, тогда он ищет библиотеку под названием `libodbc.so`, а вы можете установить символическую ссылку, чтобы приравнять ее к вашей собственной библиотеке. Например, как `root`, можно запустить команды:

```
# ln -s libodbc.so.1.0.0 /usr/lib/libodbc.so
# /sbin/ldconfig
```

Последний шаг настройки - это убедиться, что Wine установлена на запуск встроенной версии файла `odbc32.dll`, путем изменения конфигурации DLL. Эта встроенная DLL обычно работает как загрузка между вызывающей программой и Unix ODBC библиотекой.

Если у вас возникли трудности, можно воспользоваться командой `WINEDEBUG=+odbc32` перед запуском Wine, чтобы проследить, что происходит. Одно предупреждение. Некоторые программы немного мошенничают и обходят библиотеку ODBC. Например, `Crystal Reports engine` отправляется в реестр проверять DSN (имя источника данных). Способ справиться с этим, описан на сайте `unixODBC`, где имеется раздел, посвященный использованию `unixODBC` с Wine.

3.3.6.2. Использование ODBC драйверов Windows

Говорят, что "родные" ODBC драйвера работают со многими типами баз данных, включая MSSQL и Oracle. На самом деле, только MSSQL неким образом может быть доступен с Линукса через приложение Winelib. Вместо того чтобы просто копировать файлы DLL, большинство ODBC драйверов требует применения установщика Windows типа, чтобы правильно настроить некоторые вещи, вроде ключей реестра.

Чтобы наладить поддержку MSSQL, вы сначала должны скачать и запустить инсталлятор mdac_typ.exe с сайта microsoft.com. Чтобы сформировать ODBC связи, вы затем должны запустить CLICONFG.EXE и ODBCAD32.EXE под Wine. Вы можете найти их в директории windows\system, после того как закончит работу mdac_typ. Сравните выход этих программ с выходом на настоящей Windows машине. Некоторые вещи, например протоколы, могут отсутствовать, так как они могут устанавливаться вместе с операционной системой. Если дело обстоит так, то вы можете скопировать недостающее в существующей инсталляции Windows вместе с любыми необходимыми значениями в реестре. Настоящая инсталляция Windows, подготовленная для работы с Wine, должна работать сама по себе не хуже, чем до подготовки. Вот список баз данных, удачно прошедших тестирование с Wine:

| DB Type | Usefulness |
|---------|------------|
| MS SQL | 100% |

Если узнаете о других успехах, пожалуйста, сообщите в подписной лист wine-devel wine-devel@winehq.org.

Глава IV. Работа с Wine

В этой главе будут описаны все аспекты запуска программы Wine, такие как базовый вызов Wine, параметры командной строки различных сопутствующих программ и так далее.

4.1. Основное использование: приложения и апплеты панели управления

Подразумеваем, что вы используете фальшивую инсталляцию Windows. Вы устанавливаете программы в Wine точно так же, как делаете это в Windows: путем запуска инсталлятора. Можете принять папку для установки по умолчанию, большинство инсталляторов используют по умолчанию "C:\Program Files", что вполне хорошо. Если инсталлятор программы спрашивает, можете создать иконки на рабочем столе и в главном меню. Если это сделано, вы сможете запускать приложения прямо щелчком мыши.

Стандартным способом деинсталляции программ является штатный деинсталлятор самой программы, обычно зарегистрированный в апплете панели управления "Add/Remove Programs". Чтобы вызвать эквивалент деинсталлятора от самой Wine, запустите в терминале программу uninstaller (она расположена в папке programs/uninstaller/ в директории с исходниками Wine):

```
$ uninstaller
```

Некоторые программы, например Internet Explorer и QuickTime, устанавливают объединенные апплеты панели управления. Вы можете вызвать панель управления Wine, запустив в терминале:

```
$ wine control
```

которая откроет окно панели управления с установленными апплетами, как в Windows.

Если приложение не устанавливает себя в меню или значок на рабочем столе, вам придется запускать его из командной строки. Если вы помните, куда вы установили программу, то примерно такая команда:

```
$ wine "c:\program files\appname\appname.exe"
```

скорее всего сделает свое дело. Путь (path) не реагирует на регистр букв, но не забудьте про двойные кавычки. Некоторые программы не всегда используют достаточно очевидные способы именования своих программ и EXE файлов, так что можно заглянуть в директорию Program Files и посмотреть, что там записано.

4.2. Как запускать Wine

Можно просто запустить команду Wine, чтобы увидеть небольшую подсказку:

```
Wine 20040405
```

```
Применение: wine ПРОГРАММА [АРГУМЕНТЫ...] Запустить программу
```

```
wine --help Показать эту помощь и выйти
```

```
wine --version Показать версию и выйти
```

Первым аргументом должно быть имя файла, который вы хотите запустить с помощью Wine. Если исполняемый файл находится в переменной окружения Path, можно просто напечатать его имя. Если же файл не входит в Path, то нужно указать полный путь к исполняемому файлу (в формате Windows, а не Юникс!). Например, дана такая переменная окружения Path:

```
Path="c:\windows;c:\windows\system;e\;e\test;f:\"
```

Вы можете запустить файл c:\windows\system\foo.exe при помощи:

```
$ wine foo.exe
```

Однако файл c:\myapps\foo.exe придется запускать командой:

```
$ wine c:\\myapps\\foo.exe
```

(обратите внимание на двойной обратный слэш!)

О подробностях запуска исполняемых файлов в текстовом режиме (CUI) смотрите [раздел](#) ниже.

4.3. Графический интерфейс Wine, подобный интерфейсу "Проводника" Windows

Если вы предпочитаете управлять вашими файлами при помощи графического менеджера, то вам стоит использовать Winefile. Это приложение Winelib поставляется с Wine и находится вместе с прочими программами Wine. Оно удобно, чтобы видеть конфигурацию ваших дисков, и для поиска файлов, плюс к тому, вы можете запускать программы прямо из Winefile. Пожалуйста, имейте в виду, что многие функции пока не реализованы.

4.4 Опции командной строки Wine

4.4.1. --help

Показывает небольшую страничку помощи.

4.4.2. --version

Показывает номер версии Wine. Полезно для проверки вашей инсталляции.

4.5. Переменные окружения

4.5.1. WINEDEBUG=[channels]

Wine небезупречен и многие Windows программы идут под Wine не без ошибок. Справедливости ради надо сказать, что множество Windows программ идут с ошибками и под настоящим Windows! Чтобы упростить людям отслеживание причин каждой ошибки, Wine предоставляет множество каналов отладки (debug channels), которыми вы можете воспользоваться.

Каждый канал отладки, будучи активирован, начинает выводить сообщения протоколирования на консоль, в которой работает Wine. Вы можете перенаправить эти сообщения в файлы и исследовать их в свободное время. Но будьте осторожны! Некоторые из каналов отладки могут записывать в протокол немислимые объемы сообщений. Среди наиболее плодотворных вредителей выделяются relay, который записывает сообщение в протокол при каждом вызове win32-функции; win, который отслеживает прохождение Windows сообщений, и, конечно, all, который является синонимом каждого отдельного существующего канала отладки. Для сложного приложения объем ваших отладочных логов легко может превысить 1 Мб и больше. Сыщик relay часто генерирует более 10 Мб логов, в зависимости от продолжительности работы программы. (Чтобы изменить содержание отчетов relay, вам придется редактировать ключ реестра RelayExclude). Ведение всех этих логов тормозит работу Wine, так что пользуйтесь WINEDEBUG только тогда, когда действительно хотите получить лог сообщения.

Внутри каждого отладочного канала, вы можете задать класс сообщений, чтобы отфильтровывать степень серьезности ошибок. Четыре класса сообщений суть: trace (след), fixme (исправить), warn (предупредить), err (ошибка).

Для включения отладочного канала, используйте форму class+channel. Для отключения канала форму class-channel. Чтобы перечислить несколько каналов в одной опции WINEDEBUG, разделяйте каналы запятыми. Например, чтобы получать сообщения класса warn в канале отладки heap, вы можете запустить Wine такой командой:

```
$ WINEDEBUG=warn+heap wine program_name
```

Если вы не укажете класс сообщения, Wine будет выдавать сообщения всех 4 классов для данного канала:

```
$ WINEDEBUG=heap wine program_name
```

Если вы захотите получать логи со всех каналов, кроме канала relay, вы можете дать примерно такую команду:

```
$ WINEDEBUG=+all,-relay wine program_name
```

Вот список всех отладочных каналов и классов в Wine. В следующих версиях каналы будут добавлены (или убавлены).

Таблица 1. Отладочные каналы

| | | | | |
|----------|------------|-------------|------------|--------------|
| accel | adpcm | advapi | animate | aspi |
| atom | avicap | avifile | bidl | bitblt |
| bitmap | cabinet | capi | caret | cdrom |
| cfgmgr32 | class | clipboard | clipping | combo |
| comboex | comm | commctrl | commdlg | computername |
| console | crtdll | crypt | curses | cursor |
| d3d | d3d_shader | d3d_surface | datetime | dc |
| ddeml | ddraw | ddraw_fps | ddraw_geom | ddraw_tex |

| | | | | |
|-------------|-------------|-------------|-------------|------------|
| debugstr | devenum | dialog | dinput | dll |
| dma | dmband | dmcompos | dmfile | dmfiledat |
| dmime | dmloader | dmscript | dmstyle | dmsynth |
| dmusic | dosfs | dosmem | dplay | dplayx |
| dphnpast | driver | dsound | dsound3d | edit |
| enhmetafile | environ | event | eventlog | exec |
| file | fixup | font | fps | g711 |
| gdi | global | glu | graphics | header |
| heap | hook | hotkey | icmp | icon |
| imagehlp | imagelist | imm | int | int21 |
| int31 | io | ipaddress | iphlpapi | jack |
| joystick | key | keyboard | listbox | listview |
| loaddll | local | mapi | mci | mcianim |
| mciavi | mcicda | mcimidi | mciwave | mdi |
| menu | menubuilder | message | metafile | midi |
| mmax | mmio | mmsys | mmtime | module |
| monthcal | mpeg3 | mpr | msacm | msdmo |
| msg | mshtml | msi | msimg32 | msisys |
| msrle32 | msvcrt | msvideo | mswsock | nativefont |
| netapi32 | netbios | nls | nonclient | ntdll |
| odbc | ole | oledlg | olerelay | opengl |
| pager | palette | pidl | powermgnt | print |
| process | profile | progress | propsheet | psapi |
| psdrv | qcap | quartz | ras | rebar |
| reg | region | relay | resource | richedit |
| rundll32 | sblaster | scroll | seh | selector |
| server | setupapi | shdocvw | shell | shlctrl |
| snmpapi | snoop | sound | static | statusbar |
| storage | stress | string | syscolor | system |
| tab | tape | tapi | task | text |
| thread | thunk | tid | timer | toolbar |
| toolhelp | tooltips | trackbar | treeview | ttydrv |
| twain | typelib | uninstaller | updown | urlmon |
| uxtheme | ver | virtual | vxd | wave |
| wc_font | win | win32 | wineboot | winecfg |
| wineconsole | wine_d3d | winevdm | wing | winhelp |
| wininet | winmm | winsock | winspool | wintab |
| wintab32 | wnet | x11drv | x11settings | xdnd |
| xrandr | xrender | xvidmode | | |

О деталях работы с отладочными каналами, читайте [The Wine Developer's Guide](#).

4.5.2. WINEDLLOVERRIDES=[DLL Overrides]

Не всегда возможно запустить приложение, используя встроенные DLL. Иногда "родные" Виндовские DLL просто лучше работают. Хотя эти замены DLL могут быть произведены при помощи winecfg, вы можете использовать переменную окружения WINEDLLOVERRIDES для их настройки.

Для примера: вы хотите, чтобы Wine использовал "родные" ole32.dll, oleaut32.dll и rpcrt4. Тогда

запускайте Wine такой командой:

```
$ WINEDLLOVERRIDES="ole32,oleaut32,rpcrt4=n" wine program_name
```

В разделе [замена DLL](#) вы найдете дополнительную информацию по этому вопросу.

4.5.3. Настройка OSS звуковых драйверов

Если у вас звуковые драйвера OSS и есть несколько устройств (например, /dev/dsp*, /dev/mixer*), вы можете выбрать какое именно вы хотите использовать, при помощи переменных окружения

```
AUDIODEV=[audio device]
```

```
MIXERDEV=[mixer device]
```

```
MIDIDEV=[MIDI device]
```

Вот пример:

```
$ AUDIODEV=/dev/dsp4 MIXERDEV=/dev/mixer1 MIDIDEV=/dev/midi3 wine program_name
```

4.6. Опции командной строки wineserver'a

Wineserver обычно стартует автоматически, как только запускается первый Wine процесс. Однако wineserver имеет несколько полезных опций командной строки, которые можно использовать, если запускать его вручную, например, используя скрипт при залогинивании пользователя, или еще как-нибудь.

4.6.1. -d<n>

Устанавливает значение ошибки, при котором wineserver начинает вывод наладочной информации в терминале. Другими словами: всякое значение, отличное от 0, вызовет специфический вывод программы wineserver.

4.6.2. -h

Выводит на дисплей сообщение, содержащее помощь по опциям командной строки wineserver'a.

4.6.3. -k[n]

Убивает текущий wineserver, при помощи сигнала n.

4.6.4. -p[n]

Этот параметр делает wineserver постоянным на n секунд. Это предотвратит мгновенное закрытие wineserver'a.

Обычно wineserver прекращает работу почти мгновенно после окончания последнего процесса Wine, использовавшего данный wineserver. Однако, в силу того, что wineserver загружает множество вещей при старте (например, данные целого реестра Windows), его последующий старт может быть весьма долгим, так что полезно удержать его от закрытия после окончания сессии Wine, сделав его постоянным.

4.6.5. -w

Этот параметр заставит новый wineserver повременить со стартом, дав время текущему закончить работу.

4.7. Настройка переменных окружения Windows/DOS

Ваша программа может нуждаться в правильной установке переменных окружения для своей нормальной работы. В этом случае устанавливать переменную окружения нужно в Linux shell, а уже Wine переведет все переменные окружения в соответствующее пространство Windows. Вот пример для оболочки bash (у других оболочек может быть другой синтаксис!):

```
export MYENVIRONMENTVAR=myenvironmentvarsetting
```

Это обеспечит доступ вашей Windows программы к переменной окружения MYENVIRONMENTVAR, когда вы запустите программу при помощи Wine. Если вы хотите установить MYENVIRONMENTVAR на постоянной основе, вы можете поместить настройку в файл /etc/profile, или также, в случае bash, в файл ~/.bashrc.

Имейте однако, в виду, из этого правила есть исключения: Если вы захотите таким способом изменить переменные окружения PATH, SYSTEM или TEMP, вы, ни в коем случае не должны действовать подобным образом, потому что измените настройку переменных окружения Юникс. Вместо этого вам следует установить их в реестр. Чтобы сделать это запустите Wine regedit, и найдите ключ

HKEY_CURRENT_USER/Environment

Теперь вы можете создавать или изменять значения переменных как вам нужно

```
"System" = "c:\\windows\\system"
```

Эта запись определит нахождение системных файлов Windows. Системная директория Windows должна располагаться ниже директории, с установками Windows. Если у вас путь (path) к Windows такой: /usr/local/wine_c/windows, то системная директория будет: /usr/local/wine_c/windows/system. Это должно быть прописано без закрывающего слэша, и вы должны иметь право записи в эту директорию.

```
"Temp" = "c:\\temp"
```

Это директория, где будут сохраняться ваши временные файлы, /usr/local/wine_c/temp из нашего предыдущего примера. И снова, никаких закрывающих слэшей и право на запись!!

```
"Path" = "c:\\windows;c:\\windows\\system;c:\\blanco"
```

Эта запись ведет себя как настройки PATH в Юникс системах. Если запустить пасьянс в Wine следующим образом:

```
$ wine sol.exe
```

то он запустится, если файл sol.exe находится в одной из директорий, включенных в Path (конечно, если файл sol.exe находится в рабочей директории, он тоже запустится). Убедитесь, что в Path всегда прописаны ваши Windows и системная директории. Для этого нужно иметь запись "c:\\windows;c:\\windows\\system".

4.8 Программы с текстовым выводом (CUI: консольный интерфейс пользователя)

Программами с текстовым выводом являются программы, вывод которых состоит из одного текста (удивительно!). По терминологии Windows они называются CUI исполняемыми файлами, в отличие от GUI (графический интерфейс пользователя) исполняемых файлов. Win32 API обеспечивает комплект API, чтобы контролировать ситуацию, которая начинается с базовых свойств (типа печати текста), простирается до функций высокого уровня (полноэкранное редактирование, поддержка цвета, движения курсора, поддержка мыши), по пути включая такие свойства, как линейное редактирование и поддержка потоков ввода.

Решая столь широкий диапазон свойств, и их современное применение в мире Юникс, Wine действует тремя различными способами для запуска консольных программ (aka CUI исполняемые файлы):

bare streams - исходные потоки

wineconsole with user backend - с пользовательским выходным буфером

wineconsole with curses backend - с обязательным выходным буфером

Эти имена звучат немного загадочно и непонятно. "bare streams" (исходные потоки) - означает, что Wine не осуществляет никаких дополнительных преобразований, а просто передает ввод с консоли Юникс на консоль Windows. Два других способа подразумевают использование специальной программы Wine (wineconsole), которая обеспечивает дополнительные возможности и средства. Следующая таблица описывает, что вы можете (и чего не можете) делать в этих трех вариантах.

Таблица 2. Основные различия в консолях

| Функция | Bare streams | Wineconsole & user backend | Wineconsole & curses backend |
|--|---|--|---|
| Как запускать (подразумеваем, что исполняемый файл называется foo.exe) | \$ wine foo.exe | \$ wineconsole -- --backend=user foo.exe | \$ wineconsole foo.exe Можно использовать в качестве опции: --backend=curses |
| Хорошая поддержка на строчно ориентированные приложения (которые выдают информацию строка за строкой) | Да | Да | Да |
| Хорошая поддержка полноэкранных CUI приложений (включая, но не ограничиваясь, использованием цвета и мыши) | Нет | Да | Да |
| Могут запускаться, даже если не запущена X11 | Да | Нет | Да |
| Реализация | Направляет стандартные Windows потоки на стандартные Юникс потоки (stdin/stdout/stderr) | Wineconsole создает новое окно, где выводится вся информация (при условии доступности библиотеки USER32 DLL) | Wineconsole использует существующую консоль Юникс (в которой идет программа) и при помощи библиотеки (n)curses получает контроль над всей поверхностью терминала для взаимодействия с пользователем |
| Известные | | | Странно ведет себя, если |

| Функция | Bare streams | Wineconsole & user backend | Wineconsole & curses backend |
|-------------|--------------|----------------------------|--|
| ограничения | | | две (или больше) Window консоли используются в одном Юникс терминале |

4.8.1 Настройка CUI исполняемых файлов

При использовании wineconsole возможны несколько настроек. Wine (как и Windows) имеет в реестре несколько опций для каждого приложения. Это позволяет пользователю, к примеру, установить размер экранного буфера, желательного для данного приложения.

На сегодняшний день, только USER backend позволяет редактировать эти опции (мы не рекомендуем вручную вносить изменения в содержание реестра). Это редактирование начинается при помощи щелчка правой кнопкой мыши в консоли (это вызывает всплывающее меню), в котором вы можете выбрать:

- **Default (По умолчанию):** это установит настройки, общие для всех приложений, которые еще не были сконфигурированы. Итак, когда приложение впервые запускается (на вашей машине и под вашим аккаунтом) в wineconsole, wineconsole унаследует эти настройки по умолчанию для этого приложения. Впоследствии приложение будет иметь собственные настройки, которые вы уже сможете изменять по своему усмотрению.
- **Properties (Свойства):** редактирует настройки приложения. Когда вы закончите редактировать, у вас спросят, хотите ли вы:
 1. Сохранить новые настройки только на время данного сеанса (когда вы в следующий раз запустите это приложение, вы не увидите своих изменений).
 2. Использовать новые настройки в данном сеансе и сохранить их на будущее, так что в следующий раз вы снова сможете ими воспользоваться.

Вот список возможных настроек и их значение:

Таблица 3. Опции настройки Wineconsole

Размер курсора: Определяет размер курсора. Возможны 3 размера: маленький (33% от высоты символа), средний (66% от высоты символа), большой (100%).

Всплывающее меню: Как уже говорилось, это всплывающее меню настройки wineconsole запускается правым щелчком мыши в окне консоли. Однако, это может стать помехой, если приложение, которое вы запустили в wineconsole, использует правый щелчок мыши в своей работе. Нажатием клавиш Ctrl или Shift, вы вводите дополнительный контроль над правым щелчком мыши, открывающим меню. Например, если вы производите правый щелчок мышью без нажатия клавиши Shift, то событие направляется в приложение; а если вы делаете правый щелчок при нажатой клавише Shift, то открывается меню.

Quick edit (быстрый выбор): Это переключатель позволяет вам выбрать, как будет интерпретирован щелчок левой кнопкой мыши. Если вы отключите переключатель, то левый щелчок будет направлен как событие в приложение. Если включите - то с помощью левой кнопки мыши сможете выделять прямоугольную область экрана для последующего копирования в буфер обмена.

История: Позволяет установить количество команд, запоминаемых консолью. Кроме того, позволяет определить порядок сохранения повторяющихся одинаковых команд (возможно, перемежаемых другими). Если хотите сохранить их все, то отключите переключатель, если хотите сохранить только последнюю, то включите переключатель.

Police: Позволяет выбрать шрифт для консоли (файл шрифта, размер, цвета шрифта и фона).

Буфер экрана (Screenbuffer) и размер окна: Консоль, что мы видим, состоит из двух различных частей. С одной стороны, это буфер экрана, который содержит все, что программа выводит на экран. С другой стороны, это окно, которое показывает заданную часть буфера. Заметьте, что окно всегда меньше (или равно) буфера экрана. Если задать размер окна, сильно меньший, чем размер буфера, то появятся полосы прокрутки, которые позволят просматривать содержимое всего экранного буфера.

Close on exit: Если переключатель нажат, то wineconsole закроется, как только приложение завершит работу. В противном случае, консоль останется открытой, пока пользователь не закроет ее вручную. Это позволяет увидеть информацию, выводимую после завершения программы.

Edition mode (стиль редактирования): При вводе команд пользователь может выбрать два стиля:

- **Стиль Emacs:** работают те же сочетания клавиш, что и в Emacs. Например, Ctrl-A вернет курсор на начало строки. Для изучения деталей смотрите руководство Emacs.
- **Стиль Win32:** это стандартные для Windows сочетания клавиш (в основном при помощи стрелок).

Глава V. Устранение неисправностей и сообщения об ошибках

5.1. Что делать, если программа не работает?

Бывают случаи, когда вы перепробовали все, чтобы заставить эту проклятую программу заработать на данной версии Wine. Не расстраивайтесь, мы всегда здесь, чтобы помочь вам... (другими словами: "Сколько заплатите?")

5.1.1. Проверьте конфигурацию своей Wine

Посмотрите вывод команды `$ wine --version`, чтобы убедиться, что у вас новейшая версия Wine. Запустите `winecfg` и посмотрите на установки, чтобы убедиться, что они выглядят нормально. Проверьте `~/.wine/dosdevices`, чтобы убедиться, что `c:` указывает куда нужно.

5.1.2. Попробуйте разные версии Windows

В некоторых случаях изменение версии Windows может помочь.

5.1.3. Используйте различные пути (paths) для запуска

Это тоже иногда помогает. Попробуйте как `wine prg.exe`, так и `wine x:\\full\\path\\to\\prg.exe`

5.1.4. Поэкспериментируйте с настройкой DLL

Запустите `WINEDEBUG=+loaddll`, чтобы выяснить, какие DLL используются, и являются ли они "родными" или встроенными. Затем убедитесь, что у вас установлены правильные "родные" файлы библиотек в вашей директории `C:\\windows\\system` и поэкспериментируйте настройками очередности загрузки DLL в командной строке, либо в файле конфигурации.

5.1.5. Проверьте рабочее окружение!

Еще идея: не может ли рабочее окружение Wine быть испорченным? Убедитесь, что все в порядке с пакетами, от которых зависит работа Wine (gcc, glibc, X libraries, OpenGL (!), ...).

5.1.6. Используйте другой режим графического интерфейса

Проинструктируйте Wine посредством конфигурационного файла использовать либо desktop mode (режим рабочего стола), либо managed mode (управляемый режим), либо уродливый "нормальный" (normal) режим. Это тоже может помочь.

5.1.7. Проверьте приложение!

Может быть, ваше приложение имеет защиту от копирования? Многие виды таких защит пока не работают под Wine. Некоторые, возможно, заработают в будущем. (Уровень CD-ROM до сих пор не полнофункционален).

Идите на сайт GameCopyWorld и попытайтесь найти хороший крэк, который позволит снять защиту от копирования. Я надеюсь, что у вас легальная копия программы, но кто знает... :-)

5.1.8. Измените окружение Wine

Запуск приложения при наличии Windows раздела может драматически отличаться от такового при отсутствии Windows раздела. Настройте Wine на действия, противоположные тому, что были обычно. Также установите DCOM98 или DCOM95. Это может быть очень полезно.

5.1.9. Переконфигурируйте Wine

Иногда процесс инсталляции Wine изменяется, и новые версии Wine считаются с этими изменениями. Это особенно вероятно, если ваша установка произведена давно. Переименуйте директорию ~/.wine с целью резервного копирования. Запустите процесс установки, рекомендованный для вашей версии Wine, чтобы создать новую конфигурацию. Для сравнения используйте старую директорию ~/.wine. Для установок Wine из исходников применяйте для конфигурации Wine скрипт tools/wineinstall от имени пользователя. Это вполне безопасная операция. Позже вы сможете удалить новую директорию ~/.wine, и переименовать обратно старую.

5.1.10. Следите за сторонней информацией

Весьма возможно, что кто-то уже делал то, что вы пытаетесь проделать сейчас. Вам могут помочь следующие ресурсы:

Просмотрите WineHQ's Application Database в поисках подсказок, относящихся к вашей программе. Если ваша конкретная версия программы отсутствует в списке, просмотрите сходные версии, возможно там достаточно информации.

Frank's Corner имеет список приложений с детальными инструкциями по их установке. Дальнейшая помощь может найтись на форумах пользователей.

Может помочь Google, если уметь им пользоваться. Полезно бывает искать в Google Groups, конкретно в группе comp.emulators.ms-windows.wine

Существует IRC канал для Wine - Freenode.net. Вы можете попасть туда, используя любой IRC клиент, например Xchat. Вот необходимые настройки:
server = irc.freenode.net, port = 6667
и channel = #winehq

Если ваша программа требует Visual Basic Runtime Environment, то вы можете загрузить его с этого сайта Микрософт.

Если вам не хватает DLL, например mfc42, можете найти их на сайте www.dll-files.com

[Рассылочный лист](#) Wine также может помочь, особенно wine-users. Лист wine-devel может быть полезен в зависимости от типа ваших затруднений. Если вы пишете в wine-devel, то готовьтесь проделать некоторую работу по диагностике проблемы. Читайте следующий раздел, чтобы узнать, как определить источник ваших трудностей.

Если все усилия напрасны, стоит просмотреть коммерческие версии Wine на предмет поддержки вашего приложения.

5.1.11. Ищите причину!

Следующим шагом, который стоит предпринять, является поиск причины проблемы. Спектр возможных причин весьма широк - от простой ошибки в конфигурации до полной неработоспособности Wine. Следующий раздел описывает, как создать файл с отчетом об ошибке, и как начать искать неисправность. Отладочные возможности Wine описываются в Wine Developers Guide (руководство для разработчиков).

5.2. Как послать отчет об ошибке.

Пожалуйста, сообщайте об ошибках и сопутствующую информацию на сайт [Wine Bugzilla](#). Пожалуйста, просматривайте базу данных Bugzilla, чтобы выяснить, не было ли уже отчетов по данной проблеме. Если таковые уже были, пожалуйста, добавляйте вашу информацию к первоначальному отчету.

5.2.1 Отчеты об ошибках

Вот несколько простых советов, как сделать ваш отчет более содержательным, и тем повысить вероятность ответа и исправления ошибки:

1. Включите как можно больше информации, относящейся к делу.

Это означает, что нам нужно больше информации, чем просто: "Мой MS Word "падает", когда бы я его не запускал. Вы не знаете, почему?". Добавьте, как минимум, следующую информацию:

- Какую версию Wine вы используете (команда: `wine --version`)
- Название вашей операционной системы, дистрибутив и его версию (например, Linux Red Hat 7.2)
- Какой компилятор был использован, его версию (команда: `gcc -v`). Если вы не компилировали Wine, название пакета, и место, откуда его взяли.
- Версия Windows, если таковая используется с Wine. Если Windows не используется, то упомяните об этом.
- Название и номер версии программы, которую вы пытаетесь запустить, а также URL, где программа скачана (если таковой есть).
- Точная команда, которой вы запускаете программу Wine. Например: `wine "C:\Program Files\Test\program.exe"`
- Конкретные шаги, при помощи которых можно повторить ошибку.
- Любая дополнительная информация, могущая, по вашему мнению, относиться к делу, такая как версия Иксов (в случае проблем с Иксами), версия libc и прочее.

2. Прогоните программу с переменной окружения WINEDEBUG. Например, `WINEDEBUG=+relay wine sol.exe`

Это приведет к выводу на консоль дополнительной информации, которая может быть полезна при отладке программы. Также это замедлит воспроизводство программы. Бывают случаи, когда ошибка, по-видимому, исчезала при использовании опции +relay. Укажите это в вашем отчете.

5.2.2. Аварийное завершение программы

Если Wine аварийно отказывает во время выполнения вашей программы, нам важно иметь информацию об этом, чтобы попытаться понять причину отказа. Отчет может занимать много места (несколько Мб), но все же лучше сохранить его в файле. Когда появится приглашение Wine-dbg>, напечатайте quit.

Вы можете попробовать +relay,+snoop вместо +relay, но, пожалуйста, помните, что +snoop довольно нестабилен и часто вызывает отказ раньше, чем просто +relay!! Отчет с отказом в коде +snoop в большинстве случаев бесполезен! Можно также подключать другие параметры, в зависимости от природы проблемы, которую вы исследуете. Для получения полного списка параметров смотрите map-страницу к Wine.

Для того чтобы вывести запись с историей событий (трассировочный лог), используйте один из следующих методов:

5.2.2.1 Простой путь

1. Этот метод позволяет даже полному новичку получить качественный трассировочный лог событий, приведших к аварийному завершению программы.

Чтобы воспользоваться этим методом, на вашем компьютере должен быть установлен язык программирования perl. Чтобы узнать, есть ли он у вас, наберите команду: which perl. Если ответ нечто вроде: /usr/bin/perl, то вы можете работать. В противном случае переходите к разделу "Трудный путь". Если вы не уверены, все-таки сделайте работу. Когда вы попытаетесь запустить скрипт, то станет весьма очевидно, если у вас не установлен perl.

2. Смените директорию на /путь-к-wine>/tools.

3. Наберите: ./bug_report.pl и следуйте указаниям.

4. Пошлите отчет на сайт [Wine Bugzilla](#). Пожалуйста, просмотрите базу данных сайта, чтобы убедиться, что ваша проблема еще не зафиксирована, прежде чем отправлять отчет. Составьте детальное описание проблемы с сопутствующей информацией. Прикрепите "Прекрасно оформленный трассировочный лог" к письму. Не вставляйте лог в текст отчета - он слишком велик. Сохраняйте полную копию трассировочного лога, на случай, если он понадобится разработчикам Wine.

5.2.2.2. Трудный путь

Чаще всего только последние 100 (или около того) строчек трассировочного лога нужны, чтобы выяснить, где программа аварийно завершилась. Для того чтобы получить эти последние 100 строк, мы должны сделать следующее:

1. Перенаправить весь вывод программы WINEDEBUG в файл.

2. Отделить последние 100 строк в другой файл при помощи команды: tail.

Это можно сделать одним из следующих способов.

Для всех шеллов (оболочек):

```
$ echo quit | WINEDEBUG=+relay wine [other_options] program_name >& filename.out; $ tail -n 100 filename.out > report_file
```

(Эта команда запишет сообщение только в файл и затем автоматически завершится. Эту команду хорошо использовать, так как Wine выдает столько отладочных сообщений, что они переполняют терминал, пожирая процессорные циклы).

tcsh и другие csh-подобные шеллы:

```
$ WINEDEBUG=+relay wine [other_options] program_name |& tee filename.out; $ tail -n 100 filename.out > report_file
```

bash и другие sh-подобные шеллы:

```
$ WINEDEBUG=+relay wine [other_options] program_name 2>&1 | tee filename.out; $ tail -n 100 filename.out > report_file
```

Файл report_file теперь содержит последние сто строк вывода отладчика включая register dump и backtrace, которые являются самыми ценными массивами информации. Пожалуйста, не удаляйте эту часть, даже если не понимаете, что она означает.

Пошлите отчет на [Wine Bugzilla](#). Необходимо прикрепить файл report_file. Вместе с относящейся к делу информацией, о том, как вы его создали. Не вставляйте текст файла в текст самого отчета об ошибке - он довольно велик и спутает отчет. Если вы правильно сделаете всю эту процедуру, ваши шансы на получение ответа с помощью довольно велики.

Пожалуйста, просматривайте базу данных Bugzilla, чтобы выяснить, не было ли уже отчетов по данной проблеме. Если таковые уже были, пожалуйста, добавляйте ваш файл report_file и другую информацию к первоначальному отчету.

Словарь

Бинарный, или двоичный файл (Binary)

Файл, пригодный к обработке машиной, форма компиляции: шестнадцатеричная (в противоположность файлу исходного кода).

Дистрибутив

Форма, в которой обычно распространяются операционные системы на CD (обычно имеется в виду Линукс). Среда Линукс может быть представлена во множестве конфигураций: дистрибутивы могут быть предназначены для игр, научных приложений, серверных операций, настольных систем, и т.д.

DLL

DLL (Dynamic Link Library), это файл, который может быть загружен и выполнен программой по ходу основной программы. В основном, DLL является внешнем хранилищем кода для программ. Обычно несколько различных программ используют один и тот же DLL, вместо того, чтобы иметь этот код в своем составе, что сильно уменьшает размер программ. Синонимом DLL является "библиотека".

Текстовый редактор

Программа для создания или изменения текстовых файлов. В Линукс множество различных текстовых редакторов, как для графического, так и для консольного режима.

Примеры графических текстовых редакторов: nedit, gedit, kedit, xemacs, gxedit.

Примеры консольных текстовых редакторов: joe, ae, emacs, vim, vi. Запускайте их в терминале командой:

```
$ editorname filename
```

Переменная окружающей среды (Environment variable)

Переменная окружающей среды - это текстовое определение, применяемое шеллом (оболочкой) для сохранения важных системных установок. В оболочке bash shell (обычно используемой в Линукс), вы можете просмотреть все переменные окружения, запустив команду:

```
$ set
```

Если вы хотите изменить переменную, то запустите команду:


```
$ export MYVARIABLE=mycontent
```

Для удаления переменной окружения:

```
$ unset MYVARIABLE
```

Git

Git - это быстрый менеджер содержания директорий, написанный первоначально для использования в больших репозиториях, таких как Linux Kernel source (исходники ядра Линукс). Смотрите главу Git в Wine Developers Guide (руководстве разработчика Wine) для получения более детальной информации.

Пакет (Package)

Сжатый файл в формате, определенном для данного дистрибутива. Содержит файлы программы, которую вы хотите установить. Пакеты обычно устанавливаются командами пакетных менеджеров: `dpkg` или `rpm`.

root (рут, суперпользователь)

root - это аккаунт системного администратора. Чтобы запускать программы от имени root'a, примените команду:

```
$ su
```

Вас спросят пароль суперпользователя вашей системы, и после этого вы сможете выполнять роль системного администратора. Аккаунт root обозначается значком приглашения:

```
#
```

В то время как аккаунт простого пользователя - значком

```
$
```

Shell (оболочка, шелл)

Инструмент, позволяющий пользователю общаться с системой. Обычно шеллы текстовые и ориентированы на командную строку. Примерами популярных шеллов являются: `bash`, `tcsh` и `ksh`. Wine подразумевает для своей установки и работы `bash`, как наиболее популярный шелл в Линукс. Шелл обычно запускается в окне терминала.

Исходные коды (исходники, Source code)

До того, как программа подвергнется компиляции, она состоит из исходного кода. То есть это первоначально написанные инструкции, сообщающие компилятору, как должна выглядеть программа после компиляции в бинарные файлы.

Терминал

Окно терминала - это обычно графическое окно, в котором запускают шелл. Если Wine просит вас открыть терминал, вы просто кликаете мышью на значке терминала на рабочем столе, и появляется большое черное (или белое) окно. Wine подразумевает, что вы используете оболочку `bash` в окне терминала. Если это не так, просто напечатайте

```
$ bash
```

в приглашении окна терминала.