Санкт-Петербургский Государственный Политехнический Университет

Институт информационных технологий и управления

Кафедра Компьютерных Систем и Программных Технологий

Отчет по лабораторной работе № 2

По дисциплине «Проектирование ОС и компонентов»

По теме «Инсталляция, применение заданных и выбранных инсталляторов»

**Работу выполнила студентка группы № 13541/3**

Шаляпин Н.С. \_\_\_\_\_\_\_\_\_\_\_

**Работу принял преподаватель:**

Душутина Е.В. \_\_\_\_\_\_\_\_\_\_\_

.

Санкт-Петербург

2017

## Программа работы

Инсталляция, применение заданных и выбранных инсталляторов.

Используются в качестве приложений: утилита, выводящая информацию о файле для для Windows и для Linux. Утилиты были разработаны в прошлом семестре.

Лабораторная работа выполнялась в ОС: Windows 7 и Ubuntu 14.04.

Windows

|  |
| --- |
| Имя узла: ADMIN-ПК  Название ОС: Microsoft Windows 7 Максимальная  Версия ОС: 6.1.7601 Service Pack 1 сборка 7601  Изготовитель ОС: Microsoft Corporation  Параметры ОС: Изолированная рабочая станция  Сборка ОС: Multiprocessor Free  Зарегистрированный владелец: Admin  Зарегистрированная организация:  Код продукта: 00426-OEM-8992662-00173  Дата установки: 02.04.2017, 14:53:54  Время загрузки системы: 27.05.2017, 21:26:27  Изготовитель системы: ASUSTeK Computer Inc.  Модель системы: K53SM  Тип системы: x64-based PC  Процессор(ы): Число процессоров - 1.  [01]: Intel64 Family 6 Model 42 Stepping 7 GenuineIntel ~2501 МГц  Версия BIOS: American Megatrends Inc. K53SM.214, 15.08.2012 |

Linux

|  |
| --- |
| nikita@nikita-K53SM:~$ uname -a  Linux nikita-K53SM 3.19.0-80-generic #88~14.04.1-Ubuntu SMP Fri Jan 13 14:54:07 UTC 2017 x86\_64 x86\_64 x86\_64 GNU/Linux |

1. **Инсталляция в Windows**

## Inno Setup Compiler

Inno Setup — бесплатный инструментарий для создания инсталляторов на базе сценариев. Среди главных особенностей продукта — работа со всеми версиями ОС Windows, расширенная поддержка 64-битных приложений, настраиваемые типы установки, встроенный препроцессор и мощный язык сценариев на основе Pascal.

Документ разделен на секции, и каждая из них отвечает за определенную задачу инсталлятора соответственно. Всего имеется два типа секций — с параметрами и «директива—значение». Inno использует «родственный» язык — [RemObjects Pascal Script](http://www.remobjects.com/ps.aspx). Сценарии отвечают за добавление новых опций, создание интерфейса, вызов файлов или библиотек, настраиваемые действия (custom actions) и т. д. В рабочей папке Inno Setup находятся примеры скриптов (также доступны в меню компилятора). По умолчанию в Inno Setup входит компилятор сценариев Compiler IDE.

Самый простой вариант создания инсталлятора — обратиться к пошаговому мастеру настройки Inno Script Studio. Здесь указывается информация о продукте, координаты программы, файл лицензии и другие данные, языки локализации, опции компилятора. Впоследствии проект доступен для компиляции, отладки либо модификации полученного сценария в окне редактор.

Редактор Inno Script Studio доступен на русском языке. Значительно проще настраивать проект последовательно, переходя по секциям, а не создавая сценарий с нуля. При добавлении нового элемента изменения вносятся в код сценария, синтаксис подсвечивается, код соответствующим образом разбит на секции ([Setup], [Types], [Components]…).

Рассмотрим подробно код написания скрипта.  
Первым шагом определим необходимые константы.

|  |
| --- |
| ; Имя приложения  #define Name "FullInfoFile"  ; Версия приложения  #define Version "0.0.1"  ; Фирма-разработчик  #define Publisher "Nikita Shalyapin"  ; Имя исполняемого модуля  #define ExeName "FullInfoFile.exe" |

Тело скрипта разделяется на секции, каждая из которых несет свое функциональное назначение. Обязательная секция **[Setup]** задает глобальные параметры работы инсталлятора и деинсталлятора.

|  |
| --- |
| [Setup]  ; Уникальный идентификатор приложения,  ;сгенерированный через Tools -> Generate GUID  AppId={{A4BEA521-F1E2-4167-8019-0CF1ECDB91C8}  ; Прочая информация, отображаемая при установке  AppName={#Name}  AppVersion={#Version}  AppPublisher={#Publisher}  ; Путь установки по-умолчанию  DefaultDirName={pf}\{#Name}  ; Имя группы в меню "Пуск"  DefaultGroupName={#Name}  ; Каталог, куда будет записан собранный setup и имя исполняемого файла  OutputDir="D:\!\13541\_3.2\ОС (агентное программирование)\Nikita\Lab3\Setup"  OutputBaseFileName="FullInfoFile-setup"  ; Файл иконки  SetupIconFile="D:\!\13541\_3.2\ОС (агентное программирование)\Nikita\Lab3\Setup\icon.ico"  ; Параметры сжатия  Compression=lzma  SolidCompression=yes |

Здесь указываются AddId — уникальный идентификатор приложения (GUID), используемый для регистрации приложения в реестре Windows. Его генерируем командой Tools -> Generate GUID. Далее указывается имя приложения, под которым оно будет установлено в системе, его версию, данные фирмы разработчика.

Путь, по умолчанию предлагаемый инсталлятором для установки определяем опцией **DefaultDirName**. При этом переменная **{pf}** — это путь в каталог Program Files соответствующей разрядности. Опция **DefaultGroupName** определяет имя группы программы в меню «Пуск».

Пара опций **OutputDir** и **OutputBaseFileName** задают каталог, куда будет записан скомпилированный setup и его имя.

Последние опции в этой секции определяют алгоритм сжатия (LZMA) и указывают, что все файлы сжимаются одновременно, а не по отдельности (**SolidCompression**) что ускоряет процесс распаковки при большом количестве однотипных файлов.

Для поддержки нескольких языков существует секция **[Languages]**. При отсутствии данной секции будет использоваться английский язык.

|  |
| --- |
| [Languages]  Name: "english"; MessagesFile: "compiler:Default.isl"  Name: "russian"; MessagesFile: "compiler:Languages\Russian.isl" |

Каждая строка в данной секции задает один из используемых при установке языков. Параметр Name — «имя» языка, допускаются общепринятые двухбуквенные сокращения («en», «ru», «de» и так далее). Параметр MessagesFile сообщает компилятору в каком месте взять шаблон сообщений, выводимых при инсталляции. Эти шаблоны берем в каталоге компилятора Inno Setup, о чем мы сообщаем директивой compiler. Для английского языка годится шаблон Default.isl, для русского — Languages\Russian.isl

Обычно установщик предлагает нам, например, определится, хотим мы или не хотим создать ярлык на рабочем столе. Такие опции установки определяются необязательной секцией **[Tasks].**

|  |
| --- |
| [Tasks]  ; Создание иконки на рабочем столе  Name: "desktopicon"; Description: "{cm:CreateDesktopIcon}"; GroupDescription: "{cm:AdditionalIcons}"; Flags: unchecked |

Здесь Name задает имя операции — «desktopicon» — создание иконки на рабочем столе; Description — описание чекбокса с опцией, которое увидит пользователь. Конструкция задает стандартный текст сообщения, соответствующий выбранному в начале инсталляции языку. Параметр GroupDescription — заголовок группы чекбоксов с опциями. Параметр Flags задает определенные действия и состояния элементов управления, в данном случае указывая, что галочка «создать ярлык на рабочем столе» должна быть снята.

Теперь укажем, какие файлы надо включить в дистрибутив и где их надо поместить при установке. Для этого используется обязательная секция **[Files]**.

|  |
| --- |
| [Files]  ; Исполняемый файл  Source: "D:\!\13541\_3.2\ОС\Nikita\Lab3\Setup\FullInfoFile2.exe"; DestDir: "{app}"; Flags: ignoreversion  Source: "D:\!\13541\_3.2\ОС\Nikita\Lab3\Setup\icon.ico"; DestDir: "{app}"; Flags: ignoreversion |

Source — путь к файлу-источнику. DestDir — каталог установки, переменная {app} содержит путь, выбранный пользователем в окне установщика. Flags — флаги. В нашем примере для исполняемого файла: игнорирование версии программы при перезаписи исполняемого модуля, если он уже существует в системе (ignoreversion).

Наконец укажем компилятору, где брать иконки для размещения в меню программ и на рабочем столе.

|  |
| --- |
| [Icons]  Name: "{group}\{#Name}"; Filename: "{app}\{#ExeName}"; IconFilename: "{app}\icon.ico"  Name: "{commondesktop}\{#Name}"; Filename: "{app}\{#ExeName}"; Tasks: desktopicon; IconFilename: "{app}\icon.ico" |

Жмем Ctrl + F9 для компиляции инсталлятора в Inno Setup Compiler (использовалась Unicode версия). После успешной сборки инсталлятор можно запустить, нажав F9. Сначала выбираем язык, потом папку установки.

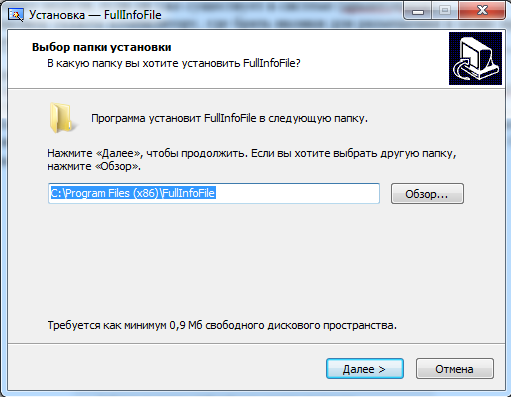


Рис. 1. Установка FullInfoFile. Выбор папки установки

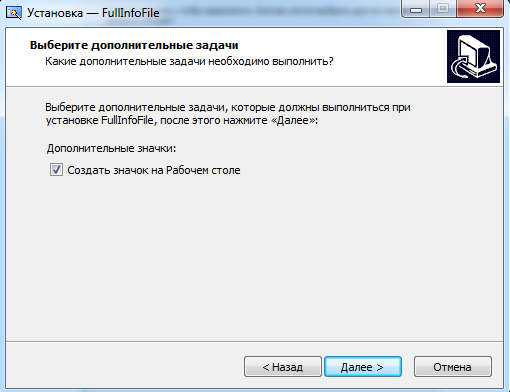


Рис. 2. Установка FullInfoFile. Дополнительные задачи – создание значка на рабочем столе

Параметры установки выбраны, нажимаем “установить”.

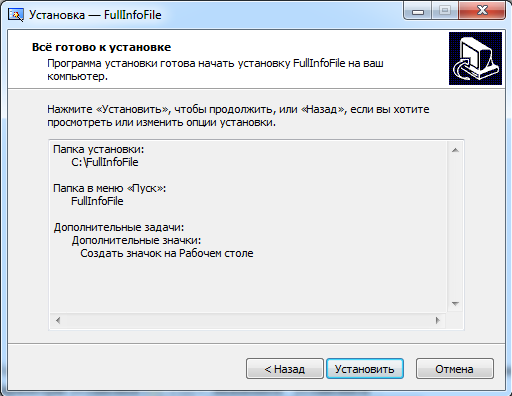


Рис. 3. Параметры установки выбраны

Установка успешно завершена.

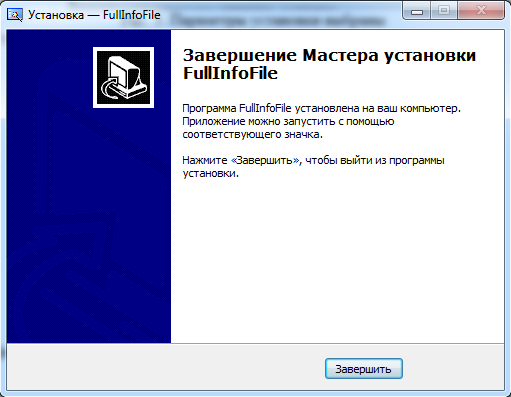


Рис. 4. Установка успешно завершена

Просмотрим папку приложения “FullInfoFile”.

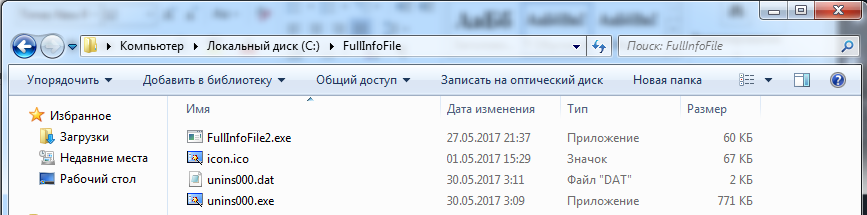


Рис. 5. Содержимое папки FullInfoFile

Ярлык программы “FullInfoFile” появился на рабочем столе и в меню “Пуск”.

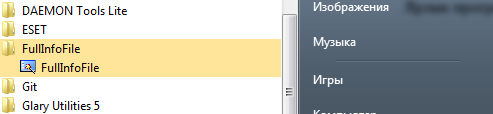


Рис. 6. Ярлыки

Программа работает корректно, далее продемонстрированно получение получение информации о файле " D:\getFileInfo.exe".

|  |
| --- |
| File Index : 295211048576  Number Of Links : 1  File Type: FILE\_TYPE\_DISK  The final path is: \\?\D:\getFileInfo.exe  The full path name is: D:\getFileInfo.exe  File attributes:  The attribute FILE\_ATTRIBUTE\_ARCHIVE is established  File size= 77404656 bytes  File creation time: 02/06/2017 18:23  File last access time: 02/06/2017 18:23  File last write time: 09/14/2016 18:04  FILE\_ENCRYPTABLE  Section: .text  =======================  Virtual size: 0xf244  Raw size: 0xf400  Virtual address: 0x1000  Raw address: 0x400  Characteristics: R X  Section: .itext  =======================  Virtual size: 0xf64  Raw size: 0x1000  Virtual address: 0x11000  Raw address: 0xf800  Characteristics: R X  Section: .data  =======================  Virtual size: 0xc88  Raw size: 0xe00  Virtual address: 0x12000  Raw address: 0x10800  Characteristics: R W  Section: .bss  =======================  Virtual size: 0x56bc  Raw size: 0  Virtual address: 0x13000  Raw address: 0x11600  Characteristics: R W  Section: .idata  =======================  Virtual size: 0xe04  Raw size: 0x1000  Virtual address: 0x19000  Raw address: 0x11600  Characteristics: R W  Section: .tls  =======================  Virtual size: 0x8  Raw size: 0  Virtual address: 0x1a000  Raw address: 0x12600  Characteristics: R W  Section: .rdata  =======================  Virtual size: 0x18  Raw size: 0x200  Virtual address: 0x1b000  Raw address: 0x12600  Characteristics: R  Section: .rsrc  =======================  Virtual size: 0xb200  Raw size: 0xb200  Virtual address: 0x1c000  Raw address: 0x12800  Characteristics: R  SCS\_32BIT\_BINARY  Для продолжения нажмите любую клавишу . . . |

## Excelsior Installer

Excelsior Installer — бесплатная программа, предназначенная для создания инсталляторов и обновлений. Основные особенности — простая интуитивная графическая оболочка.

Создание дистрибутива или обновления производится с помощью упаковщика (Packager). Он содержит пошаговый мастер, содержащий настройки проекта. На основе проекта легко скомпилировать инсталлятор.

Упаковщик состоит из разделов Files (Файлы), Settings (Настройки) и Appearance (Внешний вид). Он содержит стандартный перечень опций, отвечающих за создание лицензионного соглашения, выбор инсталляционной директории, ярлыков, ключей реестра и другие параметры.

Инсталлятор поддерживает локализацию, причем русский язык доступен «из коробки», остальное скачивается отдельно. Возможна автоматическая — тихая — инсталляция без запроса данных у пользователя.

Также существует родственный продукт — Excelsior Delivery, являющийся коммерческим. По большей части функции Excelsior Installer дублируются. Но есть также и несколько дополнительных функций: добавление в систему OLE- и ActiveX-элементов, файловых ассоциаций, библиотек, создание опросов пользователей. Лицензионная версия Excelsior Delivery (стоимостью $99) позволяет также применить к установщику фирменное оформление.

Рассмотрим по пунктам настройку инсталлятора. Первым шагом необходимо выбрать файлы, которые необходимы будут инсталлятору.

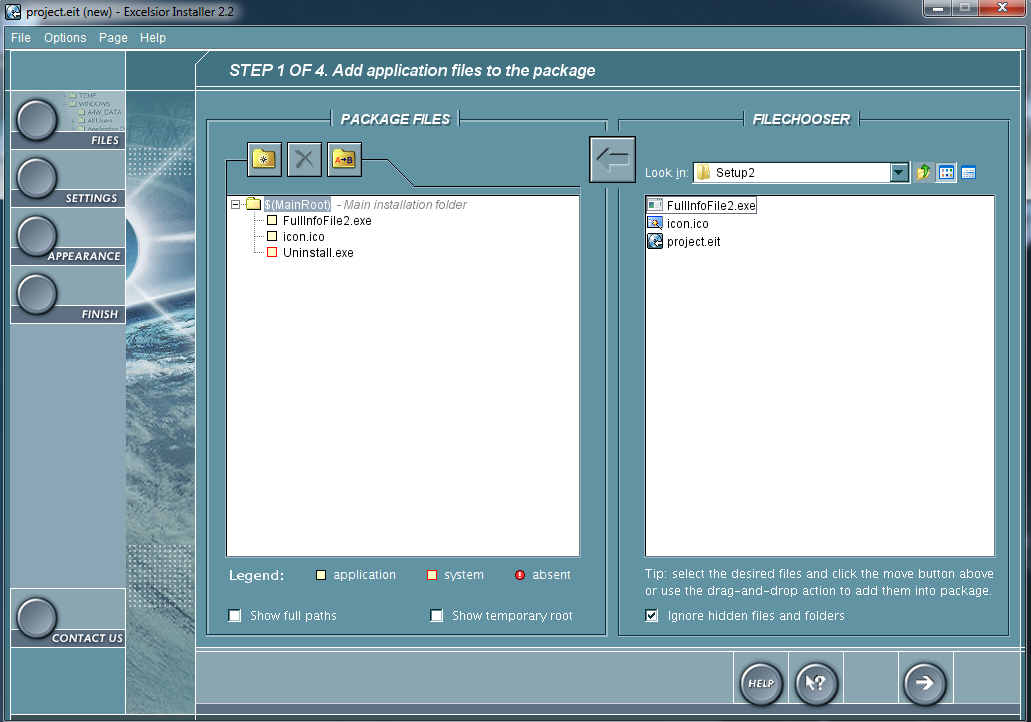


Рис. 7. Выбираем необходимые файлы

Добавим описание: имя компании, название продукта, директорию по умолчанию.

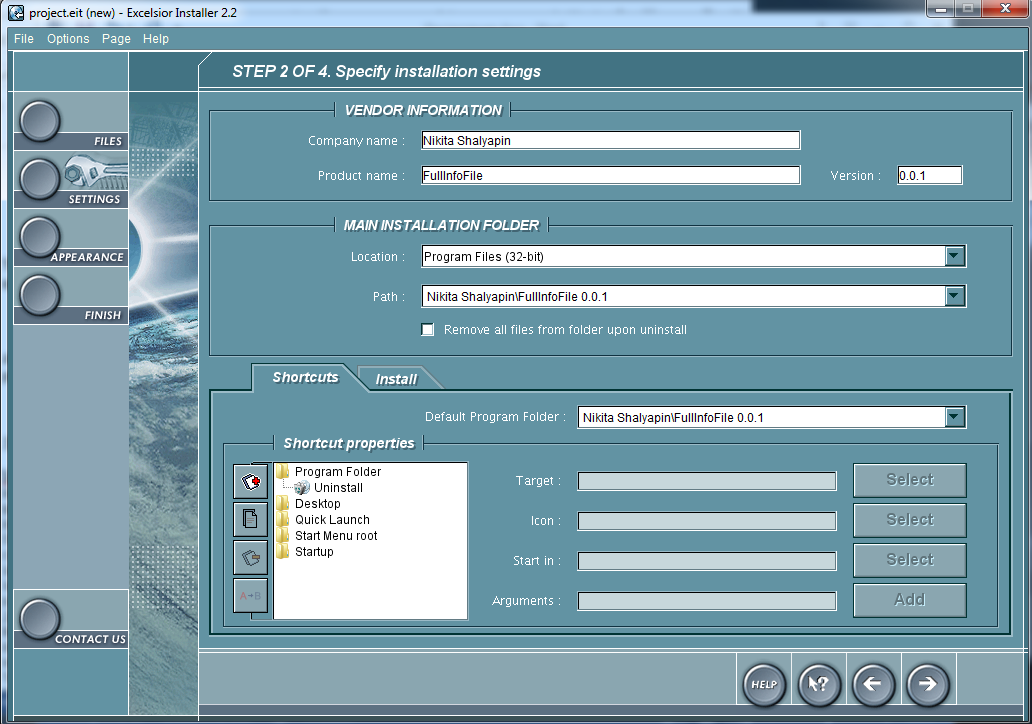


Рис. 8. Добавляем описание

Определим язык и добавим лицензионное соглашение.

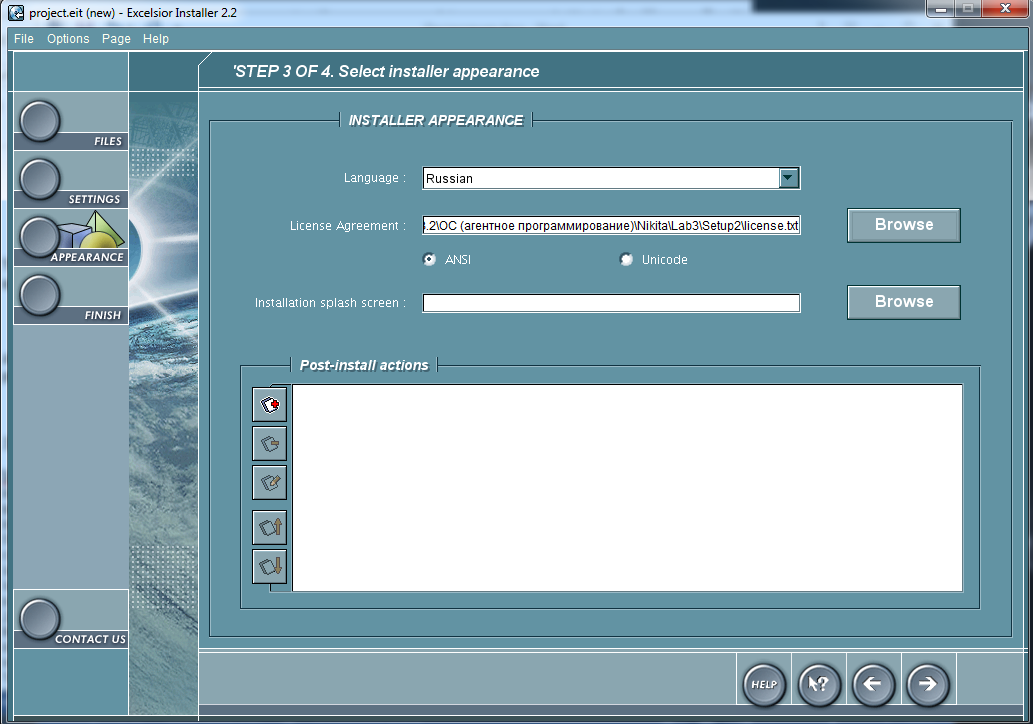


Рис. 9. Указываем язык и добавляем лицензионное соглашение

Зададим название файла инсталлятора.

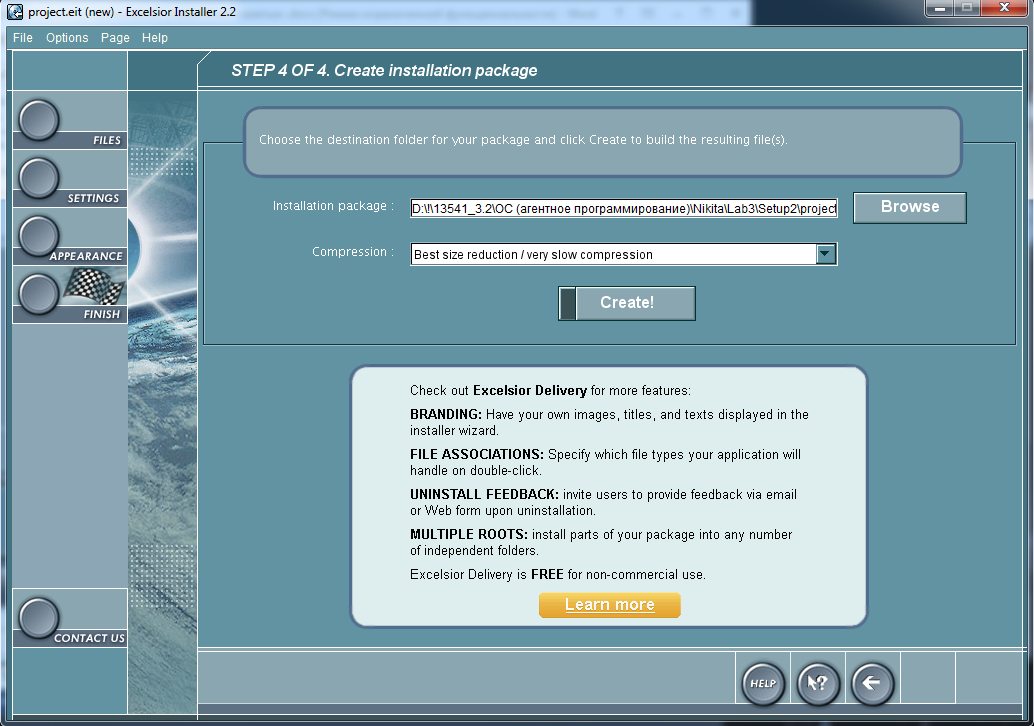


Рис. 10. Задаём название инсталлятора

Инсталлятор успешно создан.

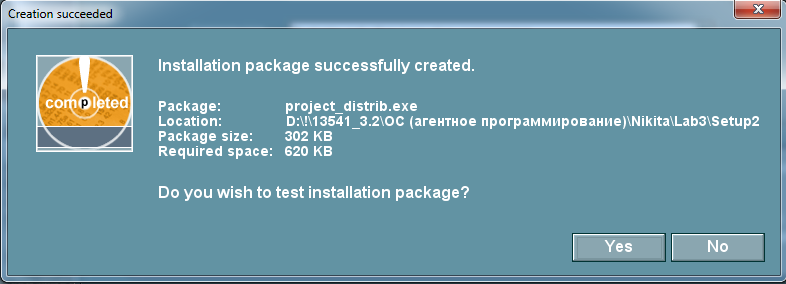


Рис. 11. Инсталлятор успешно создан

Запустим полученный инсталлятор.

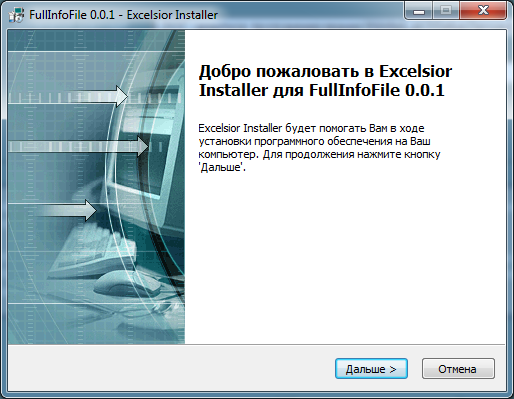


Рис. 12. Запуск инсталлятора

Заметим, что язык инсталлятора русский.

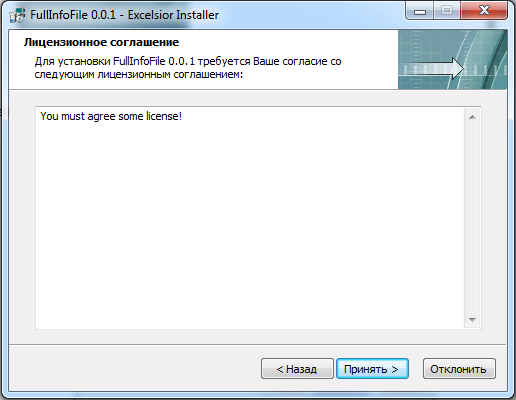


Рис. 13. Лицензионное соглашение

Далее выбор каталога установки.

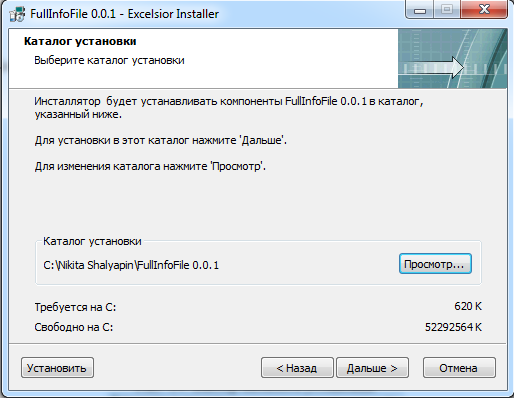


Рис. 14. Выбор каталога установки

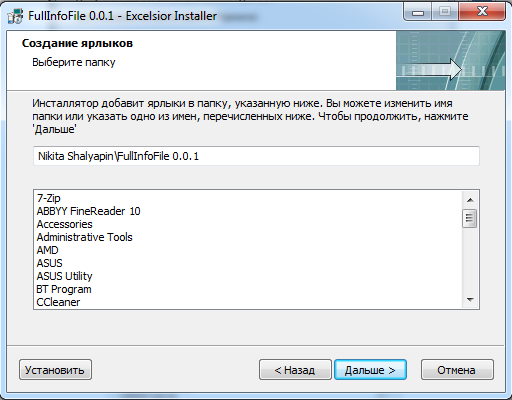


Рис. 15. Создание ярлыков

После нажимаем на кнопку “Установить”.

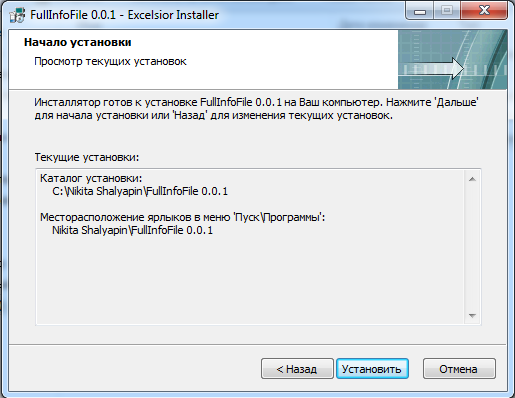


Рис. 16. Нажимаем на кнопку “Установить”

После установки просмотрим содержимое папки приложения.

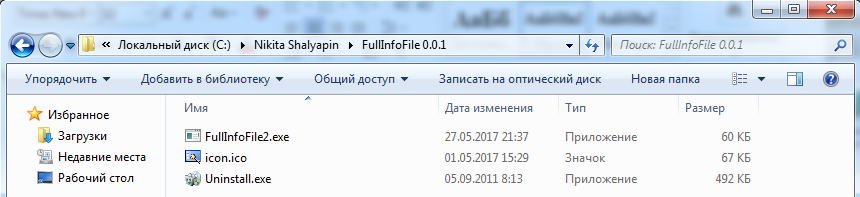


Рис. 18. Содержимое папки приложения

Ярлык программы “FullInfoFile” появился на рабочем столе и в меню “Пуск”. Программа работает корректно.

1. **Инсталляция в Linux**

В данной работе рассматривался процесс создания .deb пакета для Ubuntu 16.04. Формат .deb пакетов позволяет устанавливать в систему различные программы и библиотеки, как в виде бинарных файлов, так и из исходных файлов. Рассмотрим процесс создания пакета с бинарным файлом.

Создадим новый каталог и заполним его следующим содержимым:

|  |  |
| --- | --- |
| |  | | --- | | nikita@nikita-K53SM:~/labs/2 semester/lab2/infofile\_1.0$ tree  .  ├── DEBIAN  │ ├── control  │ └── copyright  └── usr  └── local  └── bin  └── infofile  4 directories, 3 files | |

Рассмотрим этот каталог более подробно. Имя самого каталога – infofile\_1.0 это имя пакета + версия. В нем содержится два подкаталога: DEBIAN и usr. В каталоге DEBIAN содержатся файлы с описанием пакета. Каталог usr содержит в себе устанавливаемые бинарные файлы, они расположены так же, как будут расположены после установки в систему. В данном случае мы устанавливаем единственный бинарный файл infofile, который будет лежать в /usr/local/bin/.

Файл DEBIAN/control – это файл с описанием пакета. Данный файл обязателен. Рассмотрим его содержимое:

|  |  |
| --- | --- |
| |  | | --- | | Package: infofile  Version: 1.0  Architecture: amd64  Section: misc  Maintainer: Nikita Shalyapin <cooluocker@yandex.ru>  Description: Print information about file!  .  You must pass filename as argument. | |

Рассмотрим его содержимое:

* Package – имя пакета, которое будет использоваться менеджером пакетов
* Version – версия программы и пакета, используется для обновлений пакетов и разрешения зависимостей
* Architecture – архитектура, под которую был собран бинарный файл. Допустимые значения: i386, amd64, all, source. all используется для скриптов. source используется для компилируемых пакетов с исходниками
* Sectoin – тип задачи, для которой используется приложение.
* Description – описание пакета

Так же здесь можно указывать: зависимости пакета, размер пакета и различная дополнительная информация.

Так же можно добавлять дополнительные файлы для пакета:

1. copyright – файл с описанием лицензии
2. changelog – история изменений
3. rules – правила компиляции если в пакете исходные файлы
4. conffiles – список файлов конфигурации
5. dirs – список создаваемых директорий
6. menu – файл с описанием пунктов меню
7. md5sums – контрольные суммы файлов
8. watch – мониторинг сайта программы

Создать пакет и установить его можно будет с помощью утилиты dpkg:

|  |  |
| --- | --- |
| |  | | --- | | nikita@nikita-K53SM:~/labs/2 semester/lab2$ dpkg-deb --build infofile\_1.0/  dpkg-deb: сборка пакета «infofile» в файл «infofile\_1.0.deb».  nikita@nikita-K53SM:~/labs/2 semester/lab2$ sudo dpkg --install infofile\_1.0.deb  [sudo] пароль для masha:  Выбор ранее не выбранного пакета infofile.  (Чтение базы данных … на данный момент установлено 303918 файлов и каталогов.)  Подготовка к распаковке infofile\_1.0.deb …  Распаковывается infofile (1.0) …  Настраивается пакет infofile (1.0) …  nikita@nikita-K53SM:~/labs/2 semester/lab2$ infofile test  Размер файла:  4096 байт  Число жёстких ссылок:  4  inode файла:  793083  ID устройства, содержащего файл:  2055  ID пользователя владельца:  1000  ID группы владельца:  1000  ID устройства (если специальный файл): 0  Размер блока для I/O:  4096  Число выделенных блоков размером 512 байт:  8  Права доступа к файлу:  drwxrwxr-x  Тип файла:  директория  Последнее изменение статуса:  Wed May 3 00:44:06 2017  Последний доступ к файлу:  Wed May 3 00:52:22 2017  Последния модификация файла:  Wed May 3 00:44:06 2017 | |

Как мы видим, утилита установлена и нормально работает. Удаление происходит так же с помощью менеджера пакетов.

**Дополнение**

* 1. **Утилита make**

**make**— утилита, предназначенная для автоматизации преобразования файлов из одной формы в другую. Правила преобразования задаются в скрипте с именем Makefile, который должен находиться в корне рабочей директории проекта. Сам скрипт состоит из набора правил, которые в свою очередь описываются:  
1) целями (то, что данное правило делает);  
2) реквизитами (то, что необходимо для выполнения правила и получения целей);  
3) командами (выполняющими данные преобразования).

В общем виде синтаксис makefile можно представить так:

<цели>: <реквизиты>

<команда *#1>*

...

<команда *#n>*

Для компиляции утилиты, выводящей информацию о файле, достаточно очень простого Makefile:

|  |
| --- |
| infofile: main.c functions.c  gcc -o infofile main.c functions.c |

Данный Makefile состоит из одного правила, которое в свою очередь состоит из цели — «infofile», реквизитов — «main.c functions.c», и команды — «gcc -o infofile main.c functions.c». Теперь, для компиляции достаточно дать команду make в рабочем каталоге. По умолчанию make станет выполнять самое первое правило, если цель выполнения не была явно указана при вызове:

nikita@nikita-K53SM:~/labs/2semester/lab2/infofile$ make

gcc -o infofile main.c functions.c

**Фиктивные цели**

В качестве make целей могут выступать не только реальные файлы. Командой make производят компиляцию программы, командой make install — установку. Такой подход весьма удобен, поскольку все необходимое для сборки и развертывания приложения в целевой системе включено в один файл. Обратим внимание на то, что в первом случае не указывается цель, а во втором целью является не создание файла install, а процесс установки приложения в систему. Проделывать эти действия позволяют так называемые фиктивные (phony) цели. Вот краткий список стандартных целей:

* all — является стандартной целью по умолчанию. При вызове make ее можно явно не указывать.
* clean — очистить каталог от всех файлов, полученных в результате компиляции.
* install — произвести инсталляцию
* uninstall — и деинсталляцию соответственно.

Для того чтобы make не искал файлы с такими именами, их следует определить в Makefile, при помощи директивы .PHONY. Далее показан пример Makefile с целями all, clean, install и uninstall:

.PHONY: all clean install uninstall

all: infofile

clean:

rm -rf infofile

infofile: main.c functions.c

gcc -o infofile main.c functions.c

install:

install ./infofile /usr/local/bin

uninstall:

rm -rf /usr/local/bin/infofile

Теперь мы можем собрать нашу программу, произвести ее инсталляцию/деинсталляцию, а также очистить рабочий каталог, используя для этого стандартные make цели.

|  |
| --- |
| nikita@nikita-K53SM:~/labs/2semester/lab2/infofile$ sudo make install  [sudo] пароль для nikita:  install ./infofile /usr/local/bin  nikita@nikita-K53SM:~/labs/2semester/lab2/infofile$ ls -l /usr/local/bin | grep infofile  -rwxr-xr-x 1 root root 13144 май 5 21:28 infofile  nikita@nikita-K53SM:~/labs/2semester/lab2/infofile$ sudo make uninstall  rm -rf /usr/local/bin/infofile |

* 1. **Создание .deb пакета с помощью CheckInstall**

CheckInstall — это удобная утилита, позволяющая создавать бинарные пакеты для Linux из исходного кода приложения. CheckInstall работает следующим образом. Вы собираете приложение, как делаете это обычно, но последней командой вместо make install говорите checkinstall. Программа отслеживает все изменения в системе, производимые во время установки, и создает бинарный пакет, производящий соответствующие изменения. Само собой разумеется, CheckInstall позволяет указать номер версии пакета, зависимости и тп. Поддерживаются бинарные пакеты Debian, Red Hat и Slackware.

**nikita@nikita-K53SM:~/labs/2semester/lab2/infofile$ sudo checkinstall**

checkinstall 1.6.2, Copyright 2009 Felipe Eduardo Sanchez Diaz Duran

Эта программа распространяется на условиях GNU GPL

**// Изначально утилита проверяет наличие каталога с документацией и, если таковой отсутствует, то создает его.**

The package documentation directory ./doc-pak does not exist.

Should I create a default set of package docs? [y]: y

Готовится документация к пакету...OK

\*\*\* No known documentation files were found. The new package

\*\*\* won't include a documentation directory.

**// Далее утилита просит ввести описание создаваемого пакета:**

Пожалуйста напишите описание пакета.

Закончите ваше описание пустой строкой или EOF.

>> Print information about file.

>>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* Debian package creation selected \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Этот пакет был создан с использованием данных значений:

0 - Maintainer: [ root@masha-Aspire-E5-573G ]

1 - Summary: [ Print information about file. ]

2 - Name: [ infofile ]

3 - Version: [ 20170505 ]

4 - Release: [ 1 ]

5 - License: [ GPL ]

6 - Group: [ checkinstall ]

7 - Architecture: [ amd64 ]

8 - Source location: [ infofile ]

9 - Alternate source location: [ ]

10 - Requires: [ ]

11 - Provides: [ infofile ]

12 - Conflicts: [ ]

13 - Replaces: [ ]

**// Существует возможность изменения настроек**

Введите номер для изменения параметра или нажмите ВВОД для продолжения: 3

Введите новую версию:

>> 1.0

Этот пакет был создан с использованием данных значений:

0 - Maintainer: [ root@masha-Aspire-E5-573G ]

1 - Summary: [ Print information about file. ]

2 - Name: [ infofile ]

3 - Version: [ 1.0 ]

4 - Release: [ 1 ]

5 - License: [ GPL ]

6 - Group: [ checkinstall ]

7 - Architecture: [ amd64 ]

8 - Source location: [ infofile ]

9 - Alternate source location: [ ]

10 - Requires: [ ]

11 - Provides: [ infofile ]

12 - Conflicts: [ ]

13 - Replaces: [ ]

**//После того, как были выставлены все настройки – можно начать процесс сборки пакета:**

Введите номер для изменения параметра или нажмите ВВОД для продолжения:

Installing with make install...

========================= Результаты установки ===========================

install ./infofile /usr/local/bin

======================== Установка успешно завершена ======================

Файлы копируются во временный каталог...OK

Stripping ELF binaries and libraries...OK

Сжимаются страницы руководства...OK

Построение списка файлов...OK

Собирается Debian-пакет...OK

Устанавливается Debian-пакет...OK

Удаляются временные файлы...OK

Удаляется временный каталог...OK

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Done. The new package has been installed and saved to

/home/masha/labs/2semester/lab2/infofile/infofile\_1.0-1\_amd64.deb

You can remove it from your system anytime using:

dpkg -r infofile

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**//Проверка работы утилиты, выводящей информацию о файле**

nikita@nikita-K53SM:~/labs/2semester/lab2/infofile$ cd ..

nikita@nikita-K53SM:~/labs/2semester/lab2$ infofile log.linux.txt

Размер файла:

5345 байт

Число жёстких ссылок:

1

inode файла:

793080

ID устройства, содержащего файл:

2055

ID пользователя владельца:

1000

ID группы владельца:

1000

ID устройства (если специальный файл): 0

Размер блока для I/O:

4096

Число выделенных блоков размером 512 байт:

16

Права доступа к файлу:

-rw-rw-r--

Тип файла:

обычный файл

Последнее изменение статуса:

Fri May 5 21:37:06 2017

Последний доступ к файлу:

Fri May 5 21:37:06 2017

Последния модификация файла:

Fri May 5 21:37:06 2017

**//Удаление пакета infofile**

nikita@nikita-K53SM:~/labs/2semester/lab2$ **sudo dpkg -r infofile**

(Чтение базы данных … на данный момент установлено 363476 файлов и каталогов.)

Удаляется infofile (1.0-1) …

dpkg: предупреждение: при удалении infofile каталог «/usr/local/bin» не пуст, поэтому не удалён

nikita@nikita-K53SM:~/labs/2semester/lab2$ ls -l /usr/local/bin | grep infofile

nikita@nikita-K53SM:~/labs/2semester/lab2$ dpkg -l | grep infofile

nikita@nikita-K53SM:~/labs/2semester/lab2$

Созданный deb-пакет можно использовать повторно, например, на другой машине с Debian Linux и аналогичной архитектурой процессора.

RPM обозначает две сущности: формат пакетов программного обеспечения и программа, созданная для управления этими пакетами. Программа позволяет устанавливать, удалять и обновлять программное обеспечение. RPM является основным форматом пакетов в LSB. Изначально разработанный компанией Red Hat для Red Hat Linux, RPM стал использоваться во многих дистрибутивах Linux и был портирован на другие операционные системы: Novell NetWare (с версии 6.5 SP3), IBM AIX (с версии 5).

С помощью утилиты alien его можно конвертировать, например, в RPM (права суперпользователя нужны для установки прав доступа к архивируемым файлам):

sudo alien --to-rpm infofile\_1.0-1\_amd64.deb

**Выводы**

В данной работе были изучены возможности создания инсталляторов в ОС Windows и Linux. Среди основных отличий инсталлятора Excelsior Installer от Inno Setup является то, что

* Inno Setup – проект с открытым исходным кодом, Excelsior Installer – с закрытым.
* Excelsior Installer – настройка параметров инсталлятора производится с помощью графической оболочки; Inno Setup поддерживает язык сценариев на основе Pascal.

Далее были изучены основы создания установочных пакетов для Ubuntu. Изученная система позволяет создавать пакеты как из бинарных файлов, так и компилировать эти бинарные файлы из исходных файлов во время установки.

**Список использованных источников**

1. <http://www.ixbt.com/soft/installers-1.shtml>
2. <https://www.kymoto.org/products/inno-script-studio>
3. <http://www.jrsoftware.org/isdl.php>
4. <https://habrahabr.ru/post/255807/>
5. <https://www.excelsior-usa.com/#products>
6. <http://vscode.ru/articles/sozdanie-installyatora-v-visual-studio.html>
7. <http://learn.flexerasoftware.com/content/IS-EVAL-InstallShield-Limited-Edition-Visual-Studio>
8. <https://wiki.debian.org/ru/dpkg>

**Дополнение**

**Теоретический обзор**

**Инсталляция** — процесс установки программного обеспечения на компьютер конечного пользователя. Выполняется особой программой (пакетным менеджером), присутствующей в операционной системе (например, RPM, APT или dpkg в Linux, установщик Windows в Microsoft Windows), или же входящим в состав самого программного обеспечения средством установки. В операционной системе GNU очень распространено использование системы GNU toolchain и её аналогов для компиляции программного обеспечения непосредственно перед установкой.

Инсталлятор в Windows, при инсталляции копирует файлы чаще всего в папку "Programm Files", прописывает необходимые ей данные в системном реестре, и иногда помещает дополнительные библиотеки или системные файлы в саму операционную систему.

**Реестр Windows, или системный реестр** — иерархически построенная база данных параметров и настроек в большинстве операционных систем Microsoft Windows. Реестр содержит информацию и настройки для аппаратного обеспечения, программного обеспечения, профилей пользователей, предустановки. Большинство изменений в панели управления, ассоциации файлов, системные политики, список установленного ПО фиксируются в реестре.

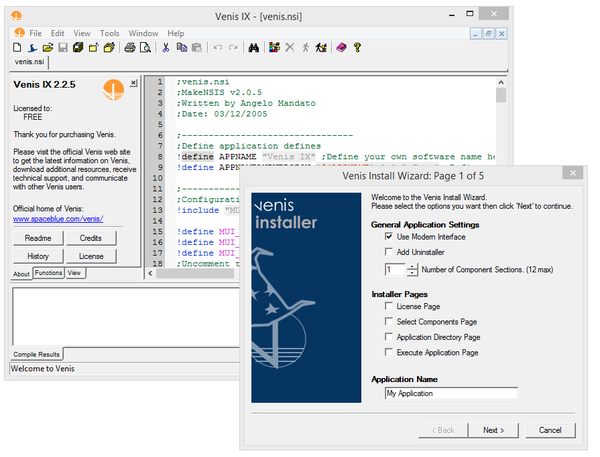
**Установщик Windows** является не только программой установки; это система управления расширяемым программным обеспечением. Установщик Windows управляет установкой программного обеспечения, управляет добавлением и удалением компонентов программ, следит за улучшением файлов и поддерживает базовое аварийное восстановление с помощью отката. Кроме того установщик Windows поддерживает установку и запуск программного обеспечения из нескольких источников и может быть настроен разработчиками, которые хотят установить настраиваемые программы.

### NSIS (Nullsoft Scriptable Install System)

NSIS — бесплатный и широко известный инструментарий для создания инсталляционных пакетов в среде Windows. Отличительные особенности — компактность, масштабируемость, поддержка плагинов и локализаций. Из других возможностей NSIS: поддержка сценариев, различных вариантов установки (включая несколько проектов внутри одного инсталлятора), создание веб-инсталляторов.

Дистрибутивы NSIS создаются на основе сценариев. Переменные, функции полностью контролируют как инсталляцию, так и деинсталляцию продукта. С помощью скриптов можно добавлять файлы и директории, вносить изменения в реестр, редактировать текстовые и двоичные файлы, создавать патчи, управлять переменными среды и даже обращаться к Windows API (посредством расширений).

Синтаксис языка NSIS достаточно прост и удобен для чтения, основная документация к нему доступна [здесь](http://nsis.sourceforge.net/Docs/). Более того, в комплекте с программой содержится большое количество примеров (файлы формата nsi доступны в папке Examples). Для редактирования nsi-сценариев пригоден любой редактор кода, где без особых усилий можно включить подсветку синтаксиса. Множество задач сразу унифицируется, если использовать специальный плагин для IDE Eclipse, [EclipseNSIS](http://eclipsensis.sourceforge.net/index.shtml), либо [Venis](http://www.spaceblue.com/products/venis/). Venis представляет собой интуитивную визуальную среду для создания сценариев.



Удобно работать с NSIS в связке и с другими программами, например, [Excelsior Installer](http://installer.excelsior-usa.com/ru/).

Если возникает необходимость расширить возможности инсталлятора, можно добавить определенные функции. Плагины пишутся на C, C++, Delphi или других языках. Готовые примеры расширений доступны [здесь](http://nsis.sourceforge.net/Category:Plugins). Функции разнообразны: от визуальных эффектов инсталлятора до создания патчей и обращения к Windows API.

Скомпилировать инсталлятор можно с помощью инструмента makensis, предварительно выбрав тип компрессии — ZLib, BZip2 или LZMA. Кроме того, сам по себе инсталлятор занимает немного места — всего 34 КБ.

Графический интерфейс инсталлятора доступен в одном из двух вариантах исполнения — Classic или Modern. Более функциональный Modern (на данный момент используется его версия 2.0) позволяет создавать страницы мастера установки с использованием текста, шрифтов, изображений, диалогами, опциями и др. На данном этапе помощь окажет программа [NSIS Dialog Designer](http://coolsoft.altervista.org/en/nsisdialogdesigner). Она позволяет быстро конструировать интерфейс, а полученный результат сохранять в формат nsdinc, который затем несложно интегрировать в .nsi-скрипт.

**Резюме**. Среда NSIS уже давно зарекомендовала себя. Несложный язык сценариев, наличие всех необходимых функций, возможность расширения, доступность основных и дополнительных инструментов — за счет всего этого NSIS составляет сильную конкуренцию программам в данной категории.

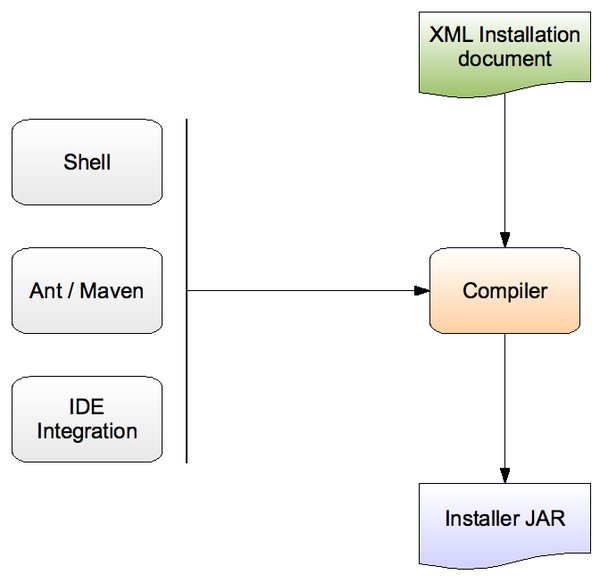
### IzPack

Домашняя страница: <http://izpack.org/>

Чаще всего, среда разработки инсталлятора закреплена за определенной платформой, и разработчик вынужден использовать различный инструментарий для каждой из них. IzPack решает вопрос кроссплатформенности, используя для своих нужд универсальную среду разработки Java.

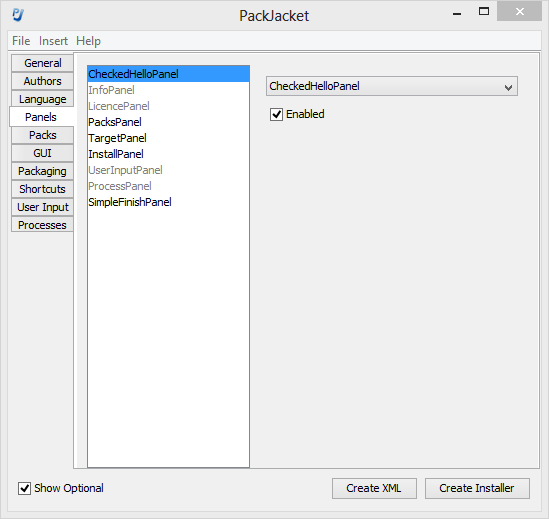
Главные особенности IzPack: XML-синтаксис, интеграция с различными инструментами компиляции, простой и гибкий API-интерфейс, возможность локализации.

Среди особенностей инсталляторов, созданных средствами IzPack, — небольшой размер и модульность. В основе дистрибутива лежит XML-документ, где указываются все необходимые спецификации, определяется функциональность продукта. Впоследствии конфигурационный файл обрабатывается любым удобным способом, для этого используется командная строка или [Apache Ant](http://ant.apache.org/). Для наращивания функциональности инсталлятора могут быть задействованы IDE среды, такие как Eclipse, IntelliJ IDEA или Netbeans. Для составления конфигурационного файла подойдет любой редактор кода.

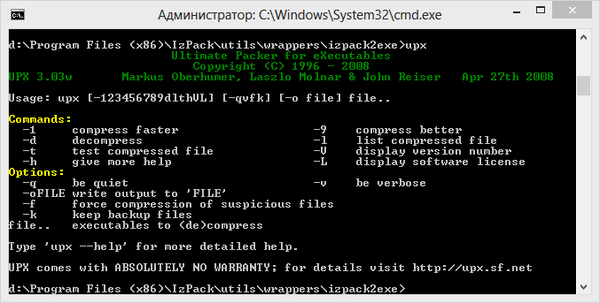


В XML-документе указывается информация о проекте (<info>), интерфейс (<guiprefs>), локализации (<locale>), файлы (<files>), ресурсы (<resources>), панели (<panels>), содержимое (<packs> — может быть упаковано в raw, gzip, bzip2) и др. Что касается java, в IzPack есть возможность применения действий (custom actions) и переменных — встроенных, пользовательских и переменных среды (см.[документацию](http://izpack.org/documentation/) и [неофициальное руководство](http://www.imrantariq.com/blog/wp-content/uploads/2011/01/izpack-installer-guide.pdf)).

Для создания графического интерфейса инсталлятора предназначены панели. С их помощью определяется внешний вид, экрана приветствия, содержимое, текстовая информация (readme, лицензии), изображения. Поддерживаются различные типы интерактивных элементов (поля, переключатели и т. д.), локализация, включая русскую. Переводу поддается как интерфейс, так и содержание.



Для создания самораспаковывающихся модулей применяется метод 7-Zip SFX, использующий возможности известного архиватора. Для Mac OS также доступна оболочка под названием izpack2app.Кроме того, есть возможность создания веб-инсталлятора, для этих целей присутствует оболочка izpack2jnlp, использующая возможности Java Web Start. IzPack поддерживает опцию тихой установки.



**Резюме**. IzPack — одна из немногих кроссплатформенных систем для создания инсталляторов. Процесс написания конфигурации, компиляции и отладки удобен за счет интеграции со средами разработки, несложен в освоении. Из недостатков следует отметить значительно устаревший (и не оригинальный — native) вид получаемого инсталлятора.

### WiX Toolset

Домашняя страница: <http://wixtoolset.org/>

WiX (Windows Installer XML) — бесплатный набор инструментов для создания инсталляторов (Windows Installer) с использованием спецификации XML. Из особенностей: интерфейс командной строки, интеграция с IDE, автоматизация процессов, поддержка базовых и расширенная поддержка Windows Installer.

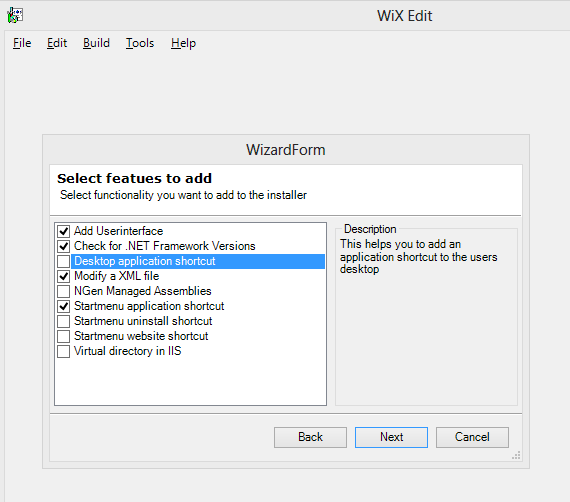
Ядро WiX составляют компилятор (candle), компоновщик (light), библиотекарь (lit), декомпилятор (dark), бутстраппер инсталляции (burn) и другие утилиты. С их помощью можно создавать пакеты инсталляции .msi, модули слияния .msm, патчей .msp.

Создание инсталлятора состоит из нескольких основных этапов: разработка сценария, его проверка, обработка препроцессором, компиляция и компоновка. Для сборки инсталлятора и автоматизации в целом используется платформа MSBuild. Автоматизация — одно из главных преимуществ WiX при работе с достаточно ресурсоемкими проектами.

Исходный код имеет формат XML (расширение файла wxs). Сценарий описывает все этапы установки, начиная с описания проекта заканчивая действиями после установки. Редактировать его можно в любом редакторе с поддержкой данного формата.

WiX радует своими интеграционными возможностями: WiX Toolset предоставляет интерфейс командной строки. Для работы с проектом в Visual Studio (модуль IntelliSense) потребуется расширение [Votive](http://wixtoolset.org/documentation/manual/v3/votive/).

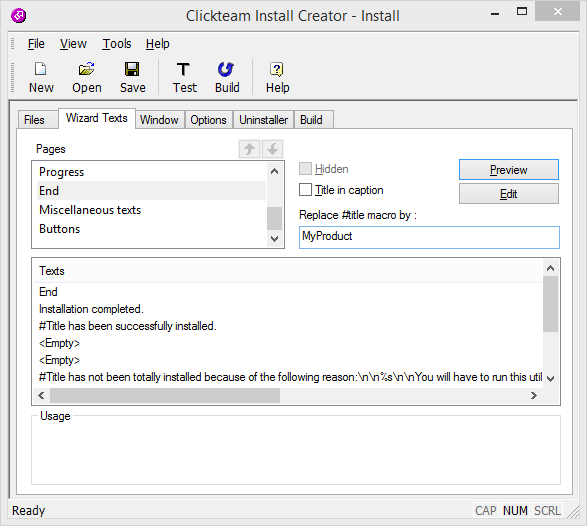
В затруднительных ситуациях или при отсутствии времени на изучение сценариев, хорошую службу окажет графическая оболочка [WiXEdit](http://wixedit.sourceforge.net/). Ее основные преимущества — импорт существующих wxs-проектов, структурирование xml-данных с удобным чтением и навигацией. С помощью мастера также несложно создать инсталляционный файл с нуля. Причем это далеко не единственный подобный инструмент (см. [здесь](http://wixtool.org/)).



**Резюме**. В данном случае, многое решает спецификация XML. WiX дает возможность программистам оперировать исходными данными, не прибегая к графической оболочке. В состав WiX входят все необходимые инструменты, вдобавок к этому функциональность инсталлятора может быть улучшена за счет расширений.

### Clickteam Install Creator 2

Домашняя страница: <http://www.clickteam.com/install-creator-2>



Компания Clickteam выпускает простые и дружелюбные продукты для разработчиков: например, Clickteam Fusion, The Games Factory, Multimedia Fusion. В этом же ряду находится программа Install Creator. Основное преимущество продукта — быстрое создание дистрибутива при помощи пошагового мастера.

Разделы настроек расположены в такой строгой последовательности в виде вкладок. Вначале указываются данные (Files), текстовое содержание диалогов (Wizard Text), формат окна (Window), дополнительные настройки (Options), настройки деинсталлятора (Uninstaller), размер дистрибутива (Build). После нажатия на Build запускается компиляция инсталлятора, остается лишь указать место его расположения.

Инсталлятор занимает 65 КБ, с учетом деинсталлятора это 105 КБ. Вдобавок к этому, в Install Creator применяется один из двух алгоритмов компрессии (zipper/ differentiator), оптимальный вариант определяется автоматически. При необходимости, данные можно поместить в пакет отдельно от инсталлятора.

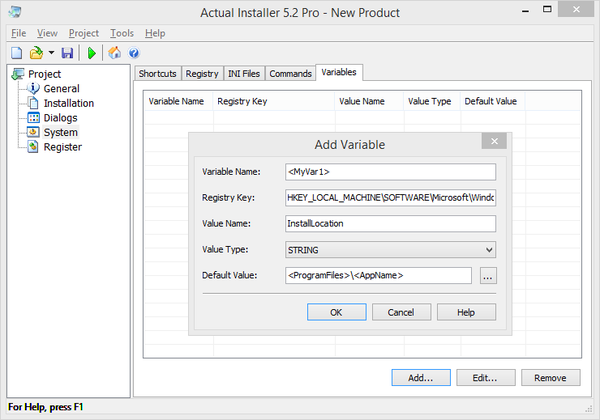
Функциональность Clickteam Install Creator унифицирована, отсутствуют некоторые важные опции. Так, мультиязычность не поддерживается, русский язык в перечне локализаций и вовсе отсутствует. Интеграция со сторонними решениями не предусмотрена, поэтому работа с большими проектами (подразумевающими множество связей в системе) будет неудобной. Программа не позволяет создавать обновления, патчи. Впрочем, в последнем случае, можно использовать [Patch Maker](http://www.clickteam.com/patch-maker) — разработку Clickteam.

В незарегистрированной версии Install Creator присутствует реклама. Профессиональная версия, помимо снятия этого лимита, позволяет включать в дистрибутив несколько редакций, устанавливать лицензионное ограничение для пользователя инсталлятора.

**Резюме**. Возможности Install Creator будут востребованы при работе с несложными проектами. Получаемые инсталляторы содержат базовый набор пользовательских опций, среди которых нет выбора языков или профиля установки. Определенно, это максимально простое решение среди всех упомянутых в обзоре инсталляционных комплексов.

### Actual Installer

Домашняя страница: <http://www.actualinstaller.com/>



Actual Installer — программа для создания достаточно сложных установщиков Windows Installer без использования сценариев. В базовый набор возможностей входят мультиязычность, эффективная компрессия, детальная настройка интерфейса, командная строка и прочие функции, перечисляемые ниже.

Использование пошагового мастера не предполагается, но доступно окно параметров для нового проекта. Определив базовые сведения продукта, можно перейти в главное окно программы. Опции расположены в 5 секциях — «Общее», «Установка», «Диалоги», «Система» и «Реестр».

В разделе General (Общее) указывается информация о проекте, здесь же настраивается внешний вид мастера установки. В процессе установщик может проверять системные требования (такие как версия ОС, привилегии пользователя) и зависимости (Microsoft .NET Framework, Internet Explorer, Adobe Reader, Java, SQL Server и прочие). При отсутствии указанных компонентов, инсталлятор потребует от пользователя скачать и установить их.

В Installation («Установка») — расположение файлов и папок, расположение исполнимого файла, настройки процессов установки, обновления и деинсталляции. Оба процесса могут осуществляться в тихом режиме, не требуя вмешательства пользователя. В довесок к установщику, может использоваться [мастер обновлений](http://www.actualinstaller.com/software-updater.html) (Actual Updater).

Dialogs («Диалоги») предоставляют доступ к «начинке» разделов, включая текст лицензионного соглашения, readme файла, в том числе в формате RTF. Здесь же — выбор языков локализации, причем русский язык числится в списке. Ненужные разделы установщика легко отключить.

System («Система») — настройка системных изменений, вносимых при инсталляции: ярлыки, ключи реестра, переменные и пользовательские команды.

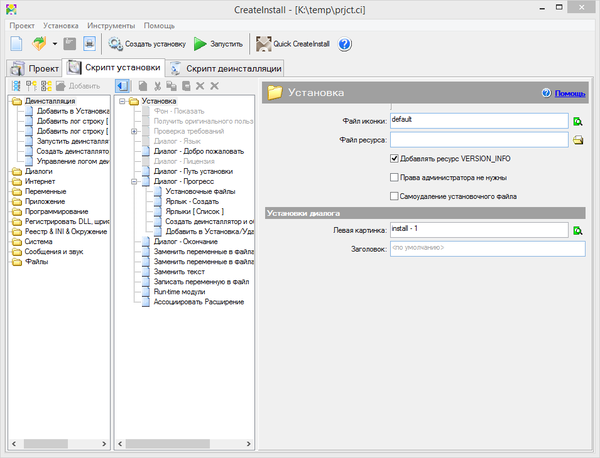
Register («Реестр») — установка файловых ассоциаций, переменных окружения, библиотек.

После выставления нужных опций достаточно нажать на кнопку Build Project — инсталлятор готов к запуску. В случае, если при создании инсталлятора была пропущена важная опция, при компиляции Actual Installer даст об этом знать и укажет на ошибку.

**Резюме**. Actual Installer включает в себя практически все стандартные параметры и опции проекта. Это «промежуточный» вариант между очень простыми (вроде Clickteam Install Creator) и более тяжеловесными решениями, о которых также пойдет речь.

### CreateInstall

Домашняя страница: <http://www.createinstall.ru/>



CreateInstall — инструментарий для создания установщиков. В его основу заложено две особенности — контроль над процессом установки и неограниченная расширяемость. Обе возможности реализованы благодаря [языку программирования Gentee](http://www.gentee.ru/), применяемому для написания сценариев.

Интерфейс CreateInstall разбит на 3 вкладки — «Проект», «Скрипт установки» и «Скрипт деинсталляции». Первый раздел позволяет задать общие настройки инсталлятора: информация о продукте, поддерживаемые языки, пути, внешний вид. Дополнительно, инсталлятор можно защитить цифровой подписью и установить пароль.

«Проект» — не равноценная замена двух последующих разделов, т. е. для создания дистрибутива нужно тщательно настроить скрипты установки и деинсталляции. Соответствующие параметры отображаются в виде групп, можно отобразить их единым списком.

Дополнением для CreateInstall служит утилита Quick CreateInstall. Она значительно упрощает создание инсталлятора, предоставляя только базовые настройки проекта. Из Quick CreateInstall в дальнейшем проект можно импортировать в CreateInstall.

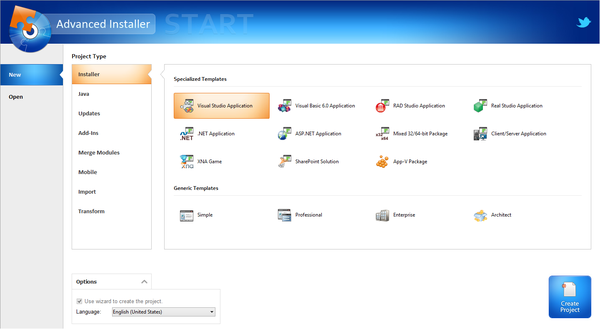
Код проекта не предназначен для самостоятельного редактирования, переноса в IDE-среду, экспорта. Хотя язык Gentee имеет отличный потенциал: как минимум, это переменные и функции, условные выражения и синтаксис, базирующийся на C, C++ и Java.

Существует 3 редакции программы — полная, light (простая) и бесплатная. Список различий между ними опубликован [здесь](http://www.createinstall.ru/help/compare-installers.html). По большей части, отсутствующие функции бесплатной редакции программы видны при навигации по группам настроек скриптов. Интерфейс и справка доступны на русском языке.

**Резюме**. CreateInstall — функциональная и доступная для понимания среда. Инструментарий CreateInstall не так уж и беден и позволяет обходиться собственными средствами. Но к недостаткам все же следует отнести замкнутость программы «в себе», подразумевая функции импорта и экспорта.

### Advanced Installer

Домашняя страница: <http://www.advancedinstaller.com/>



Advanced Installer основывается на технологии Windows Inslaller, позволяя создавать msi-, exe- и других видов дистрибутивов. Этому способствует продуманный интерфейс и работа с проектами. В Advanced Installer можно обнаружить немало возможностей, которых нет в других подобных комплексах, о чем ниже.

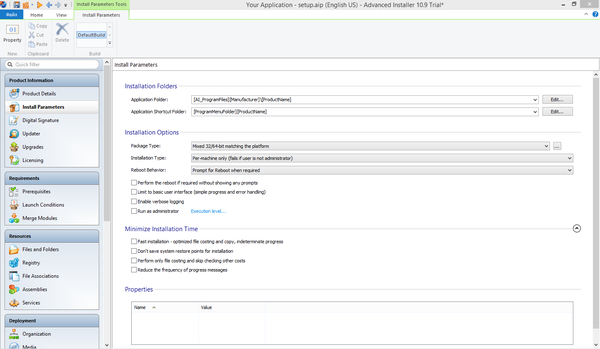
Примечательно, прежде всего, разнообразие проектов: сюда входят инсталляторы, Java-установщики, обновления, дополнения, модули слияния и другие. В разделе меню Installer собраны команды импорта проектов из Visual Studio, RAD Studio, Real Studio, Visual Basic. Здесь раскрывается потенциал Advanced Installer во взаимодействии с IDE-средами.

Для каждого из выбранных типов проекта предусмотрен детальный мастер настройки. Есть общие шаблоны — Simple, Enterprise, Architect или Professional. Большая часть проектов доступна только для определенных типов лицензии, общедоступные проекты обозначены как None в графе License Required.

Как уже сказано, при создании проекта можно воспользоваться пошаговым мастером, где, в частности, доступен выбор способа распространения пакета, языков локализации, настройка пользовательского интерфейса, ввод текста лицензии и другие опции. Advanced Installer позволяет выбрать вариант распространения программы — оставить данные без компрессии, разделить на CAB-архивы, сохранить в MSI и др., добавить цифровую подпись, потребовать ввод серийного номера и т. д.

Главное окно Advanced Installer (редактор проекта), в простом режиме отображения (Simple), содержит несколько секций:

* Product Information (Информация о продукте) — ввод сведений о продукте, параметры установки.
* Requirements (Требования) — указание аппаратных и системных требований, зависимостей ПО. Также имеется возможность создания пользовательских условий.
* Resources (Ресурсы) — редактор ресурсов (файлов и ключей реестра).
* Deployment (Развертывание) — выбор типа распространения продукта. Это может быть MSI, EXE или веб-инсталлятор. Для MSI, EXE ресурсы можно поместить отдельно от инсталлятора.
* System Changes — переменные среды.



При выборе ресурсов могут использоваться файлы, ключи реестра, переменные окружения, конфигурационные ini, драйверы, базы данных и переводы. С помощью модулей объединения можно добавить и другие ресурсы, такие как сервисы, разрешения, ассоциации и др.

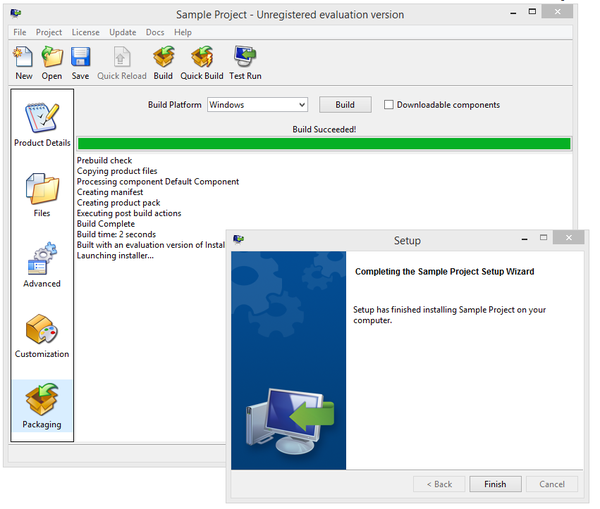
Для выполнения более сложных задач позволяется использовать пользовательские действия, EXE, DLL или скрипты, написанные на C, C++, VBS или JS. Для создания сценариев предусмотрен удобный редактор.

Однако следует отметить, что в режиме Simple доступна лишь малая часть разделов. Работая с Advanced Installer в ознакомительном режиме, есть смысл зайти в настройки и переключиться в другой режим работы с проектом. После этих действий становятся доступны новые подразделы редактора.

**Резюме**. Благодаря различным редакциям продукта (включая бесплатную), возможностям импорта, множеству шаблонов и схем, редактору проектов — Advanced Installer будет интересен широкой аудитории.

### BitRock InstallBuilder

Домашняя страница: <http://installbuilder.bitrock.com/>



InstallBuilder — кроссплатформенная среда для создания инсталляторов. В перечень поддерживаемых систем входят: Windows ME — Windows 8, Mac OS X, FreeBSD, OpenBSD, Solaris (Intel & Sparc), AIX, HP-UX, IRIX, Linux (Intel x86/x64, Itanium, s390 & PPC). Инсталляторы доступны как для настольного, так и сетевого ПО. В первой части путеводителя уже рассматривался продукт IzPack, но для InstallBuilder характерна поддержка языков написания, включая не только Java, но и PHP, Perl, Python, Ruby, C/C++ и .NET/Mono.

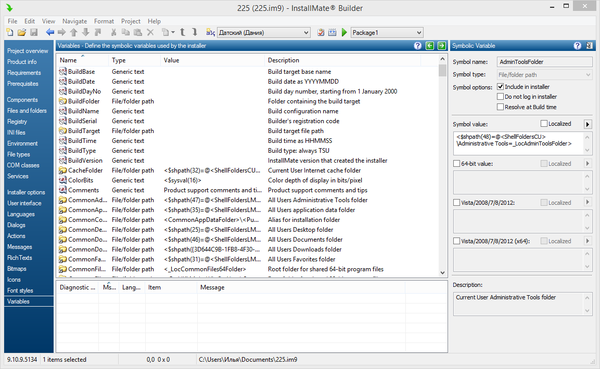
Интерфейс программы доступен на русском языке. Для работы с проектом используется очень простая графическая оболочка. Присутствуют следующие секции с настройками:

* Product Details («Информация о продукте») — информация о проекте.
* Files («Файлы») — проводниковый интерфейс для добавления файлов
* Advanced («Расширенные настройки») — наиболее интересный раздел, где, пожалуй, и раскрывается вся мощь BitRock Installer. Для работы с проектами используется формат XML (см. обзор WiX в первой части), возможна совместная работа, ручная настройка проектов с поддержкой сценариев. Встроенные действия позволяют автоматизировать процессы установки и объединять задачи, делая их доступными в несколько кликов. Таким образом можно проверять зависимости, добавлять переменные окружения, изменять реестр и файловую систему и прочее. Для редактирования сценариев используется встроенный, очень простой, XML-редактор.
* Customization («Настройка») — настройка пользовательского интерфейса инсталлятора, переменных окружения, пост-установочных действий, платформ, разрешений и компрессии. Для сжатия применяется ZIP/LZMA-компрессия. Можно создать тихий установщик, работающий в текстовом режиме.
* Packaging («Упаковка») — для отслеживания ошибок имеется встроенный дебаггер. При построении легко создать кроссплатформенный инсталлятор. На выходе он использует «родную» среду ОС (это относится к графическим системам Windows, KDE и Gnome).

**Резюме**. BitRock InstallBuilder рекомендуется, прежде всего, для создания кроссплатформенных продуктов. Сценарий, основанный на XML-спецификациях, удобен в обращении, проект легко импортировать и экспортировать для редактирования во внешней среде. Таким образом, на основе данного XML-сценария можно быстро генерировать инсталляторы для всех требуемых платформ.

### InstallMate

Домашняя страница: <http://www.tarma.com/im9/index.htm>



InstallMate представляет собой инструментарий для создания дистрибутивов на основе Windows Installer. Основные особенности: интуитивная графическая среда, широкий выбор элементов установки, проверка зависимостей, детальная настройка внешнего вида инсталлятора, поддержка компонентов, локализаций и расширений. Дистрибутивы, созданные с помощью InstallMate, отличаются компактностью и скоростью работы.

Мастер создания нового проекта позволяет выбрать существующий шаблон и указать начальные параметры — языки локализации (один или несколько), имя продукта.

Можно импортировать созданный проект или воспользоваться редактором InstallMate. Кстати говоря возможности импорта не впечатлили: поддерживаются продукты компании Tarma и проекты Visual Basic.

Главное окно программы содержит разделы с параметрами, разделенные, условно, на три секции:

* Общие настройки: детальная информация о проекте, программные зависимости, системные требования, компоненты, а также опции инсталлятора. В боковой панели Package можно также указать тип инсталлятора, компрессии, ссылку на скачивание (в случае с веб-инсталлятором), пароль на установку. В один инсталлятор можно добавить несколько пакетов.
* Компоненты: включение в дистрибутив элементов и связей: это файлы и папки, ключи реестра, ini-конфиги, переменные среды, файловые ассоциации, COM-классы, сервисы.
* Опции инсталлятора: с помощью редактора можно отредактировать текст диалогов. Предусмотрена гибкая настройка локализаций, InstallMate поддерживает 19 языков по умолчанию. Можно создать один дистрибутив с несколькими локализациями либо несколько проектов с одним языком. В этом же разделе доступны пользовательские действия, а также переменные — включая 250 [предустановок](http://www.tarma.com/support/im9/using/symbols/variables/allvars.htm). Расширить функциональность пакета можно с помощью расширений и DLL-библиотек.

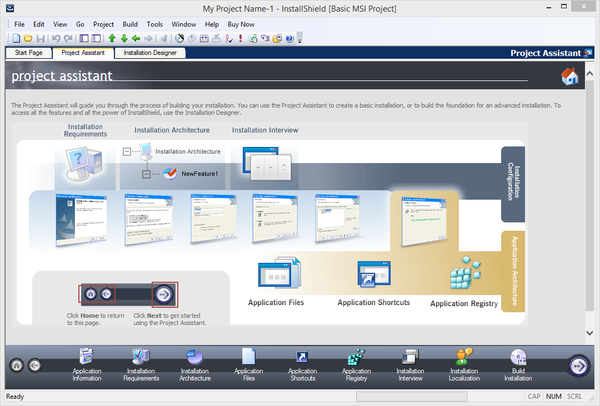
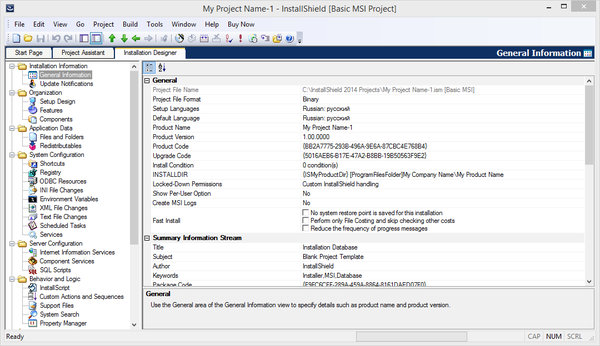
Как было отмечено, при создании инсталлятора можно использовать шаблоны. Помимо шаблонов проектов, возможно создание пакетов на основе компонентов, определяя группы элементов. Из других опций стоит отметить скрытую и тихую установку и деинсталляцию, два типа сжатия (Deflate или LZMA).

Полный список функций представлен [здесь](http://www.tarma.com/im9/features.htm).

**Резюме**. Профессиональная среда для создания инсталляторов с действительно удобным интерфейсом и быстродоступным инструментарием: можно оперативно переключиться с одного языка локализации на другой при создании дистрибутива, протестировать проект. Не хватает функционального пошагового мастера настройки, возможностей импорта. Сильная сторона — наличие всех стандартных функций, работа с переменными и действиями.

### InstallShield 2014

[Домашняя страница](http://www.flexerasoftware.com/products/installshield/editions.htm)



InstallShield предоставляет собой среду для настройки проекта, создания сценария и конечного релиза дистрибутива — как в MSI-, так и в различных виртуальных форматах. Это одно из наиболее известных решений в области создания инсталляторов для платформы Windows. Существует три редакции комплекса — Express, Professional и Premier. Последняя из них считается наиболее функциональной (см. [сравнение версий](http://media.flexerasoftware.com/documents/Datasheet-IS2014-Compare-Editions.pdf)).

Проекты могут быть импортированы из Microsoft Visual Studio или созданы на основе имеющихся шаблонов. При создании проекта с нуля на выбор доступен один из нескольких типов:

* Базовый MSI-проект — используется технология Windows Installer, доступен вызов InstallScript-, VBScript-, Jscript-скриптов, exe- и dll-файлов.
* Проект InstallScript — InstallScript используется для контроля установки
* InstallScript MSI-проект — совместное использование Windows Installer и InstallScript для развертывания установки.

свою очередь, для каждого из указанных типов имеется множество дополнительных проектов. В целом, кроме инсталляторов, комплекс позволяет создавать модули объединения (Redistributable), а также модификаторы (Transform).

Есть несколько вариантов начала работы в InstallShield, один из них — воспользоваться пошаговым мастером настройки (Project Assistent), который запускается автоматически при создании нового проекта. Более детальная конфигурация осуществляется в окне дизайнера установки. Настройки сгруппированы следующим образом:

* Информация об установке (Installation Information) — таблица с общей информацией о проекте. Здесь, в частности, можно выбрать язык инсталлятора по умолчанию. Всего InstallShield поддерживает более 30 локализаций.
* Organization (Организация) — включает в себя дизайн установщика, список функций и компонентов, которые будут представлены пользователю на выбор при установке.
* Application Data (Данные приложения) — данные приложения, такие как файлы, модули и зависимости. Часть компонентов может быть установлена локально, другие необходимо скачивать во время установки.
* System Configuration (Системная конфигурация): ярлыки, ресурсы, конфигурационные файлы, сервисы и др.
* Server Configuration — конфигурация сервера в соответствии с требованиями инсталлятора. Сюда входят настройки IIS, сервисы компонентов, SQL-скрипты для SQL Server, Microsoft Windows Azure, MySQL, или Oracle. Используя Microsoft Web Deploy, можно скачивать зависимости, запускать скрипты, выполнять задания на сервере.
* Behavior and logic (Поведение и логика) — скрипты, действия, файлы для инсталляции и др.
* User Interface (Пользовательский интерфейс) — дизайн инсталлятора, включая оформление диалогов, изображения, текст и прочие данные. На выбор доступны темы оформления, очень гибкая настройка уже имеющихся диалогов вплоть до перетаскивания каждого элемента. Построчный редактор позволяет работать с локализациями.
* Media — управление переменными окружения, настройка обновлений и релизов продукта, патчи.
* Дополнительные инструменты (Additional Tools): сканнер файловых зависимостей, MSI-отладчик и прямой редактор, предназначенный для просмотра структуры проекта в виде таблицы.

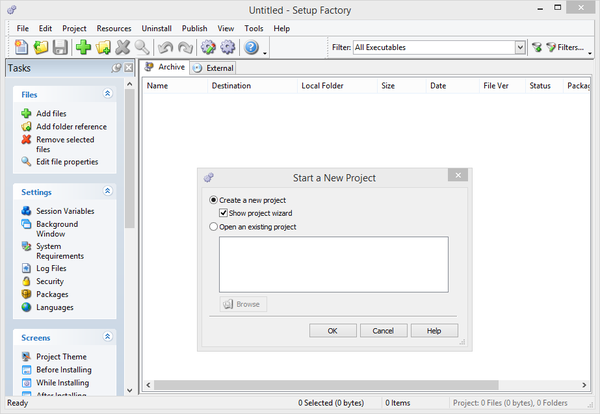
В качестве скриптового языка для создания инсталляторов выступает InstallScript, имеющий сходство с языком C и позволяющий создавать пользовательские действия (Custom Actions). Тем самым, скрипты помогают при вызове dll-библиотек, запуске приложений, VBScript-, Jscript-, InstallScript- скриптов и мн. др.

Для правки кода в InstallShield используется редактор с подсветкой синтаксиса и автодополнением. Кроме того, для создания действий может использоваться мастер пользовательских действий (Custom Action Wizard).

**Резюме**. Бесспорно, InstallShield является одним из лидирующих продуктов на рынке инсталляционных комплексов. Его наиболее сильная сторона заключается в гибкости, что позволяет создавать многочисленные типы инсталляторов для развертывания на компьютере или сервере, с поддержкой сценариев. От домашнего использования может отпугнуть достаточно высокая цена лицензии для каждой из редакций, начиная с InstallShield Express ($649).

### Setup Factory

[Домашняя страница](http://www.indigorose.com/products/setup-factory/)



Компания Indigo Rose Software Design выпускает продукты для разработки, начиная с 1991 года. Среди них — MSI Factory, TrueUpdate, Visual Patch и смежные решения. Setup Factory — еще один комплекс, позволяющий создавать полнофункциональные скриптовые инсталляторы — но, в отличие от MSI Factory, независимые от MSI. При этом полученный дистрибутив может быть запущен под любой версией Windows, начиная с XP. Доступны альтернативные варианты распространения — установка с CD- или DVD-носителя, веб-инсталлятор с возможностью скачивания зависимостей. Поддерживаются форматы компрессии LZMA/LZMA2 и файлы размером более 2 ГБ.

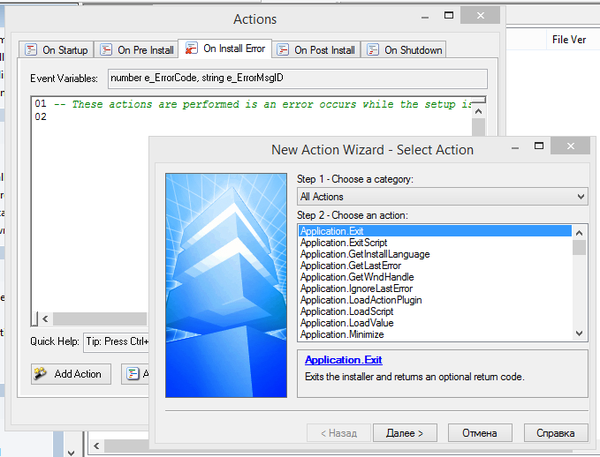
Инсталляционная среда Setup Factory использует простой, гибкий и мощный скриптовый язык, основанный на языке LUA. Это позволяет задействовать условия, функции, переменные, таблицы и другие средства при написании сценария.

Для создания инсталлятора предусмотрен традиционный пошаговый мастер. Он состоит из следующих этапов работы с проектом: заполнение общей информации об инсталляторе, выбор архитектуры, расположения файлов, стиля и темы оформления, языков установки, состав библиотек.

Как вариант, можно сразу же перейти к самостоятельной настройке проекта установщика, для этих задач предоставляется простой интерфейс редактора. В левой колонке окна приложения доступен список задач, он разбит на секции: файлы, настройки, визуальные опции, действия, ресурсы и деинсталляция.

Управление скриптами осуществляется через секцию Actions. Все действия делятся на несколько этапов, в хронологическом порядке: запуск, процесс установки и возникновение ошибок на данном этапе, окончание установки и завершение работы.

Для составления действий (actions), пользователям предоставлены надстройки Action Wizard и Code Wizard. Всего доступно около 360 скриптов, при этом, каждое действие тщательно задокументировано в хорошо структурированной справке. Параллельно с этим, основная работа с кодом ведется в окне редактора. Подсветка кода и нумерация строк в наличии, для проверки сценария на ошибки можно использовать встроенные средства Setup Factory.



Функциональность инсталлятора расширяется за счет ресурсов. В соответствующем разделе Resources можно указать зависимости (.NET, Visual Basic, DCOM, JET, MDAC, Windows Installer, Visual C++ и пр.), установить глобальные функции и подключить плагины для работы с буфером обмена, архивами, FTP. Кроме того, на форуме разработчика выложены дополнительные расширения.

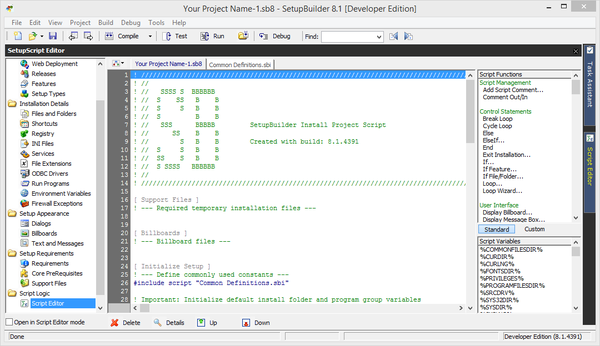
Удобство редактора проектов заметно при работе с локализациями: можно не только добавить язык интерфейса, но также импортировать или экспортировать данные в XML. Каждый аспект инсталлятора поддается настройке и доступен в режиме предосмотра: диалоги, подписи к кнопкам, форматирование текста, шрифты и мн. др. Внешний вид меняется с помощью предустановленных тем. Причем используются не только «родные» стили (Windows 7, Windows 8 и др.), но и различные MSStyles-модификации, которые можно скачать отдельно.

Напоследок стоит отметить такой аспект Setup Factory, как безопасность. Установка может быть ограничена с помощью серийного номера, все ключи хранятся в MD5-хэше. Во время инсталляции производится проверка безопасности данных, срока годности дистрибутива.

**Резюме**. Setup Factory позволяет задействовать широкие скриптовые возможности языка LUA. Все действия отлично документированы, пользователь может использовать мастер настройки на различных этапах работы, начиная созданием и заканчивая публикацией проекта. Функциональность инсталлятора может быть расширена благодаря дополнениям. Тщательно настраивается интерфейс установщика, удобна работа с локализациями.

### SetupBuilder

[Домашняя страница](http://www.setupbuilder.com/products.htm)



SetupBuilder относится к инструментам для быстрой разработки и настройки приложений. С его помощью можно создавать скриптовые инсталляторы с компактным ядром размером в 160 КБ. Это могут быть веб-, CD/DVD-дистрибутивы, работающие под любой ОС Windows 9.x и выше. Среди ключевых особенностей SetupBuilder значится сертифицированная поддержка Windows 8 и Windows Server 2012 — что гарантирует совместимость и корректность установки.

Следует отметить, что есть две редакции приложения на выбор: для разработчиков (Developer) и профессиональная версия (Professional). Перечень возможностей существенно [отличается](http://www.setupbuilder.com/products_setupbuilder_pro_features.htm) между этими редакциями, с той особенностью, что более функциональна SetupBuilder Developer. Она включает в себя графический редактор диалогов, отладчик, компилятор, генератор обновлений и прочие компоненты.

Создавая проект с нуля, можно воспользоваться мастером, однако он, по сути, мало что решает. Редактор проектов состоит из следующих разделов: Project Definition (Свойства проекта), Installation Details (Установки), Setup Appearance (Внешний вид), Setup Requirements (Зависимости) и Script Logic (Сценарий).

Project Definition — настройка общей информации, переменных, профилей установки. Следует отметить подраздел Web-deployment (Веб-развертывание), где можно активировать функции веб-обновления или веб-инсталляции продукта. Также, один проект может включать в себя несколько версий приложения (Releases).

Installation Details — определение содержимого инсталлятора, включая файлы, папки, ключи реестра, сервисы, драйверы, переменные окружения и прочие элементы.

Setup Appearance — настройка оформления. Для определения внешнего вида инсталлятора — включая диалоги, текстовое содержание, кнопки, поля и т. д. — используется визуальный редактор. Прочие настройки, такие как локализация и стиль инсталлятора, указываются отдельно в настройках проекта (Project — Settings).

Setup Requirements — проверка зависимостей, прав доступа, системных и прочих файлов.

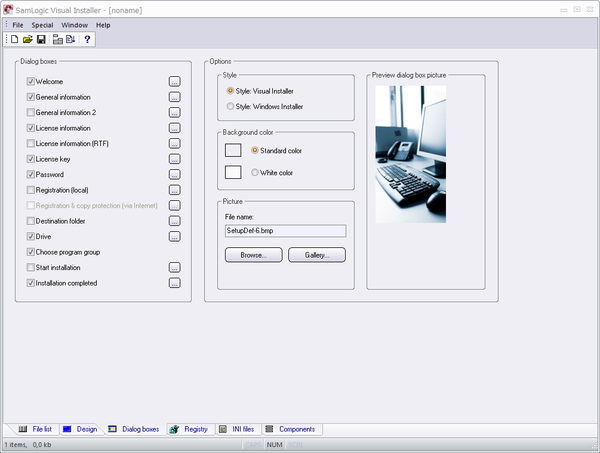
Script Logic — доступ к сценарию инсталлятора. Для работы со скриптами используется собственная IDE-среда, также имеется встроенный визуальный дебаггер. В SetupBuilder задействован скриптовый язык SetupScript, и, в случае с редакцией Developer, прилагается около 500 предустановленных функций. Кроме использования SetupScript, во время установки можно обратиться Windows API или подключить необходимые DLL-библиотеки.

За безопасность отвечает функция SetupProtect, доступная в SetupBuilder Developer. Она позволяет использовать серийные номера и ключи подписки на определенный период работы. В дополнение, доступна цифровая подпись дистрибутива.

**Резюме**. SetupBuilder позволяет создавать инсталляторы, веб-инсталляторы и обновления, используя скрипты и возможности Windows API. Из наиболее примечательных сторон данного комплекса — тесная поддержка Windows, визуальные редактирование и отладка проекта, многоязычность и защита от несанкционированного использования.

### Visual Installer 2014

[Домашняя страница](http://www.samlogic.net/visual-installer/visual-installer.htm)



Visual Installer — комплекс для создания windows-инсталляторов в визуальном режиме и при поддержке сценариев. Создаваемые дистрибутивы могут распространяться на CD/DVD-, USB-носителях и через интернет, в виде самораспаковывающихся архивов.

Есть две версии Visual Installer — стандартная и профессиональная. В последнем случае, в состав дополнительно включены менеджер проектов и SamLogic Selector — утилита для определения системных зависимостей при запуске. Кроме того, профессиональная версия позволяет работать с 64-битными приложениями и предполагает работу со сценариями.

Вначале Visual Installer предлагает указать формат распространения дистрибутива и установить базовые настройки. Как вариант, можно импортировать в программу имеющиеся проекты из сред Visual Studio / Visual Basic и продолжить работу, основываясь на существующем проекте.

Далее работа осуществляется непосредственно в редакторе. В нижней его части доступны вкладки для навигации между разделами. В разделах File list, Registry, INI Files и Components определяется структура инсталлятора, включая файлы, директории, ключи реестра и другие компоненты. Отдельно можно выделить визуальную составляющую. Так, при настройке мастера установки предоставляется выбор диалогов (вкладка Dialog Boxes): ввод ключа лицензии, выбор компонентов, текст лицензии и т. д. Дизайн установщика определяется в соответствующей вкладке Design: это цвет, фон, шрифты, изображения и звук.

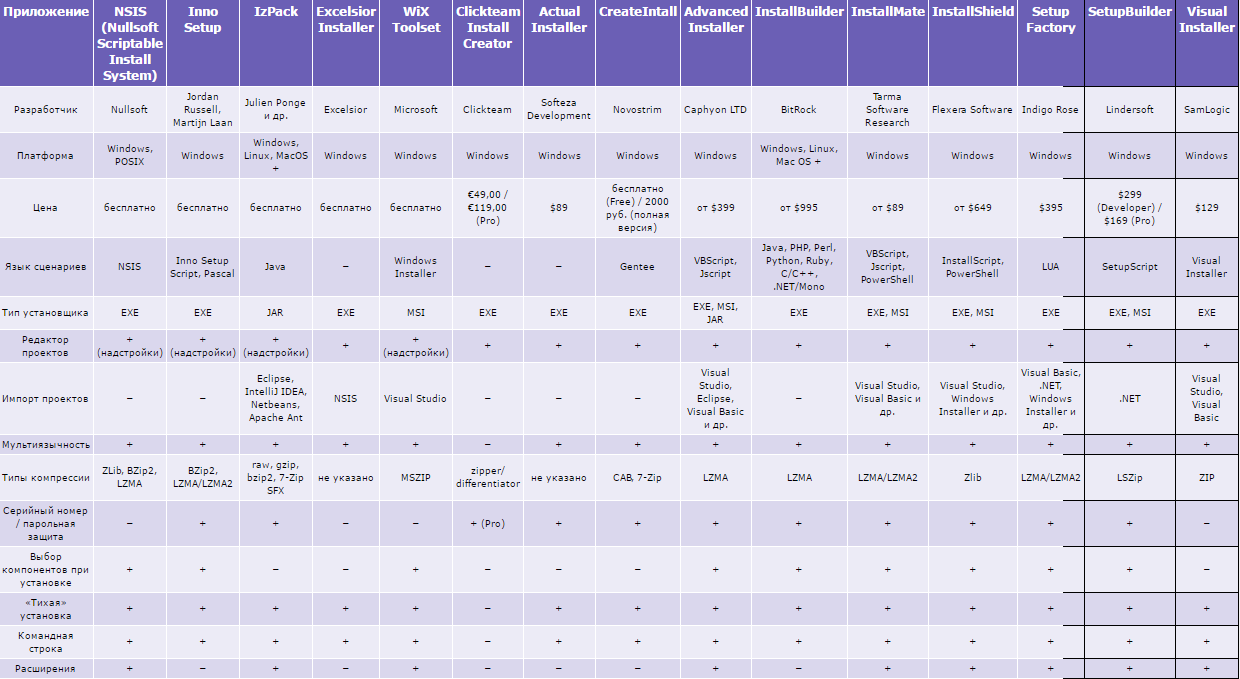
Дополнительные опции инсталлятора доступны в разделе меню Specials. Здесь можно указать зависимости (версия операционной системы и .Net Framework), без которых установка будет невозможна. Также доступны такие опции инсталлятора как тихая установка, цифровая подпись и парольная защита.

Visual Installer предусматривает создание многоязычных дистрибутивов. Управлять локализациями, по сравнению с другими инсталляционными комплексами, не очень удобно: все файлы хранятся в LNG-формате во внешнем файле, который необходимо редактировать вручную в текстовом редакторе.

Сценарии, доступные в Pro-версии, могут быть задействованы перед установкой и после ее завершения. Скриптовый язык позволяет использовать переменные и команды, описанные в справочнике, управлять процессом создания файлов, папок, ярлыков, других компонентов, запускать внешние MSI- и прочие программы. Для редактирования сценариев используется простой редактор с поддержкой автодополнения IntelliSense.

Напоследок, нужно отметить следующий момент: в диалогах некорректно отображается кодировка и не все активные кнопки срабатывают при нажатии. Данная проблема наблюдается в Windows 8.

**Резюме**. Достаточно простой и несколько устаревший инсталляционный комплекс, где, тем не менее, доступны сценарии и визуальный редактор проектов. Все настройки осуществляются на основе заготовок, изменить порядок установки нельзя.

****