

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ
1С:ПРЕДПРИЯТИЕ 8. УСТАНОВЩИК
Версия 1.0.2 (29.11.2019)
январь 2020

Пустая страница

ОГЛАВЛЕНИЕ

Особенности Установщика	1
Кроссплатформенность	1
Отсутствие привязки к операционной системе	1
Установка программного обеспечения не привязана к дистрибутиву	1
Домашняя директория программных продуктов	1
Реестр программных продуктов	1
Автоматический перезапуск	1
Возможность автоматизации установки	1
Запуск Установщика	1
Запуск графической версии	2
Запуск консольной версии	2
Использование графического интерфейса	2
Проверка цифровой подписи	4
Выбор компонентов	5
Настройки пользователей	5
Процесс установки	5
Диагностика сбоев	7
Параметры запуска	7
Использование консольного интерфейса	8
Просмотр версии и справки в консоли	8
Общие параметры	8
Команда query	8
Параметры	9
Примеры	10
Команда install	10
Опции	10
Параметры	11
Примеры	12
Команда uninstall	13
Опции	13
Параметры	14
Примеры	15
Подсистема support	15
Запуск и удаление Установщика после установки	16
Запуск установленного Установщика	16
Удаление установленного Установщика	16
Особенности поддержки различных операционных систем	17
Особенности поддержки macOS	17
Особенности поддержки Linux	17
Приложение 1. Формат файла inventory.loc	17

Приложение 2. Формат YAML-файла для установки и удаления с помощью консольной версии	17
Приложение 3. Спецификация Семантического Версионирования (SemVer)	19

ОСОБЕННОСТИ УСТАНОВЩИКА

КРОСПЛАТФОРМЕННОСТЬ

Установщик работает максимально похоже на всех поддерживаемых платформах. Так же на всех платформах доступен графический интерфейс пользователя.

ОТСУТСТВИЕ ПРИВЯЗКИ К ОПЕРАЦИОННОЙ СИСТЕМЕ

Установщик не имеет привязки к пакетным менеджерам операционных систем. Это дает возможность сделать поведение одинаковым на разных операционных системах и преодолеть их ограничения. Например, можно устанавливать несколько произвольных версий одной программы одновременно.

УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НЕ ПРИВЯЗАНА К ДИСТРИБУТИВУ

Одно и то же программное обеспечение, например 1С:Enterprise Development Tools, может в будущем входить в состав нескольких дистрибутивов. При этом не важно, сколько и каких дистрибутивов будет устанавливаться на компьютер – программное обеспечение уникальных версий будет установлено единожды без дублирования.

ДОМАШНЯЯ ДИРЕКТОРИЯ ПРОГРАММНЫХ ПРОДУКТОВ

При первой установке программного обеспечения Установщиком можно указать произвольную *домашнюю директорию* программных продуктов или использовать стандартную. Все устанавливаемое программное обеспечение в дальнейшем копируется в эту директорию без возможности изменения. Это позволяет иметь всегда одну копию нужного программного обеспечения без дублирования. При необходимости домашнюю директорию можно потом перенести на любой диск, поставив ссылку на файловой системе на первоначальную директорию.

РЕЕСТР ПРОГРАММНЫХ ПРОДУКТОВ

Все данные об установленном программном обеспечении хранятся в *реестре программных продуктов*. Это набор текстовых файлов в специальном формате, не привязанный к функциям операционной системы. Формат файлов является простым для понимания однако мы оставляем за собой право изменять этот формат произвольным образом, при этом Установщик всегда будет поддерживать старые форматы. Это позволяет Установщику хранить всю необходимую метаинформацию не вмешиваясь в ОС. Все данные хранятся в одной директории, которую при желании можно изменить в конфигурационном файле, но вряд ли это потребуется т. к. реестр программных продуктов достаточно компактный и не занимает много места.

АВТОМАТИЧЕСКИЙ ПЕРЕЗАПУСК

При установке какого-либо программного обеспечения Установщик копируется на диск. Если затем будет установлено программное обеспечение из другого дистрибутива, то Установщик либо обновится на новую версию автоматически, либо автоматически перезапустится если в состав дистрибутива включает более старую версию. Таким образом всегда работает самая актуальная версия Установщика.

ВОЗМОЖНОСТЬ АВТОМАТИЗАЦИИ УСТАНОВКИ

Установщик имеет две версии - [с графическим интерфейсом](#) и [с интерфейсом командной строки \(консольная версия\)](#). Консольный вариант Установщика можно использовать для автоматизации управления составом программного обеспечения на компьютере.

ЗАПУСК УСТАНОВЩИКА

Для работы Установщика необходимо, чтобы на компьютере была установлена 64-разрядная Java Platform, Standard Edition версии 8 или 11. Существуют разные дистрибутивы OpenJDK, но для работы Установщика важно, чтобы в составе сборки была поддержка OpenJFX(JavaFX). Мы рекомендуем использовать один из следующих дистрибутивов OpenJDK 11:

- [Liberica JDK](#);
- [Zulu Community](#).

Путь к директории с исполняемым файлом java должен быть в переменной окружения `PATH` или необходимо определить переменную окружения `JAVA_HOME`. Если необходима тонкая настройка запуска Java то необходимые параметры можно указать в переменной окружения `E1C_INSTALLER_OPTS`.

Установщик имеет две версии - консольную (для командной строки) и графическую. Консольная запускается файлом `lce-installer-cli`, а графическая — файлом `lce-installer`. В настоящее время для операционной системы Windows файлы имеют расширение CMD, но в будущих версиях скрипты планируется заменить на исполняемые файлы.

Далее необходимо распаковать архив дистрибутива (ZIP или TAR.GZ архив) и запустить необходимую версию Установщика (в Linux обязательно с правами администратора).

ЗАПУСК ГРАФИЧЕСКОЙ ВЕРСИИ

Linux

1. открыть терминал,
2. перейти в директорию с дистрибутивом и выполнить `sudo ./lce-installer`.

Windows

1. в проводнике перейти в директорию дистрибутива,
2. запустить файл `lce-installer.cmd` — запрос прав администратора будет выполнен автоматически. Для старых версий установщика может потребоваться вручную запустить скрипт от имени администратора.

macOS

1. в Finder кликнуть два раза по скачанному DMG-файлу с дистрибутивом,
2. в появившемся окне кликнуть два раза на `lce-installer` (или `lce-installer.app` если у вас отображаются расширения всех файлов) - запрос прав администратора будет выполнен автоматически.

ЗАПУСК КОНСОЛЬНОЙ ВЕРСИИ

Linux и macOS

1. открыть терминал,
2. перейти в директорию с дистрибутивом и выполнить `sudo ./lce-installer-cli`.

macOS

1. в Finder кликнуть два раза по скачанному DMG-файлу с дистрибутивов,
2. открыть терминал,
3. перейти в директорию с дистрибутивом и выполнить `sudo ./lce-installer-cli`.

Windows

1. запустить приложение **Командная строка** от имени администратора:
 - в меню программ выбрать **Командная строка**, вызвать контекстное меню и выбрать **Запуск от имени администратора**,
 - или для Windows 8.1 и более поздних версий можно вызвать меню комбинацией клавиш **Win+X** и выбрать пункт меню,
2. перейти в директорию с дистрибутивом и выполнить команду `lce-installer-cli`.

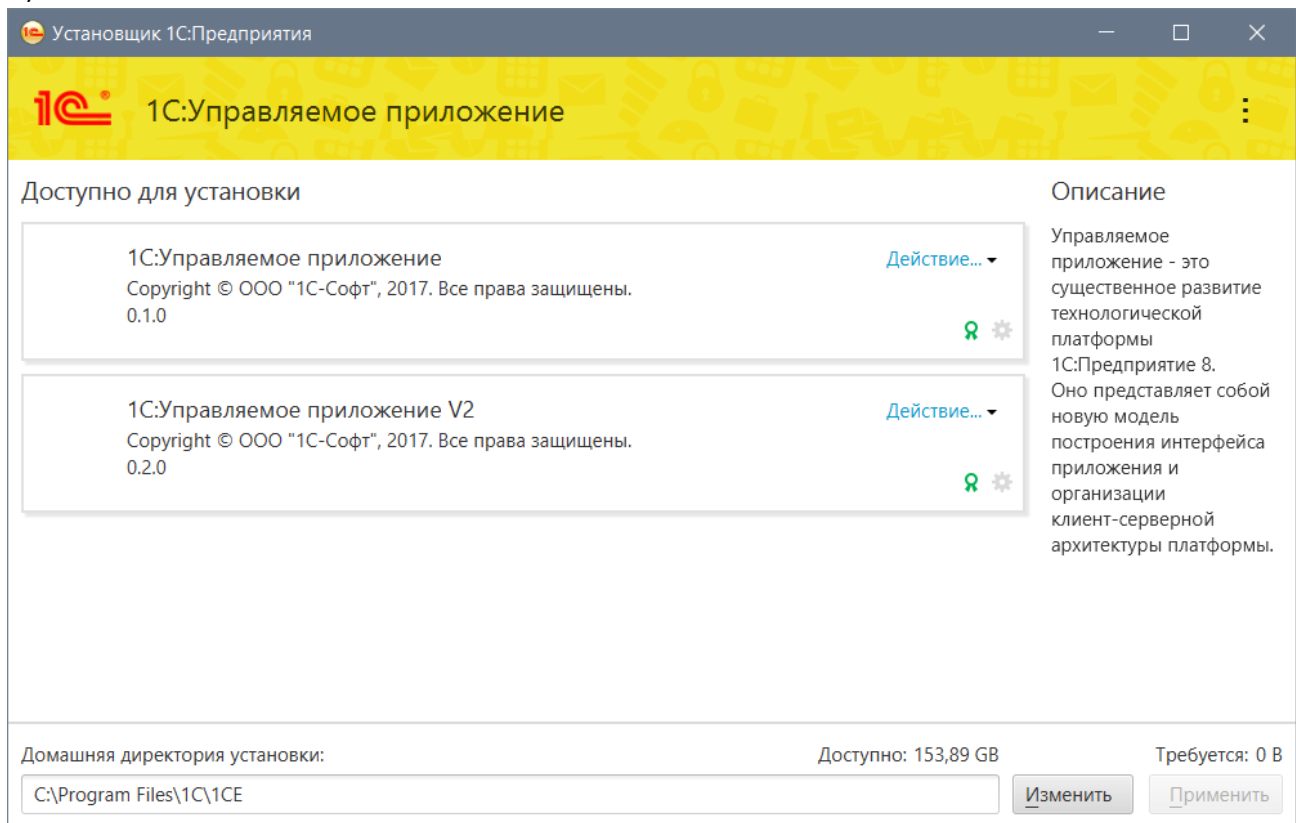
Так же можно запустить командную строку от имени обычного пользователя и воспользоваться командой `run as` (в директории дистрибутива), например:

```
runas /env /user:Administrator "cmd /c c:\path\to\unpacked\distro\lce-installer-cli.cmd"
```

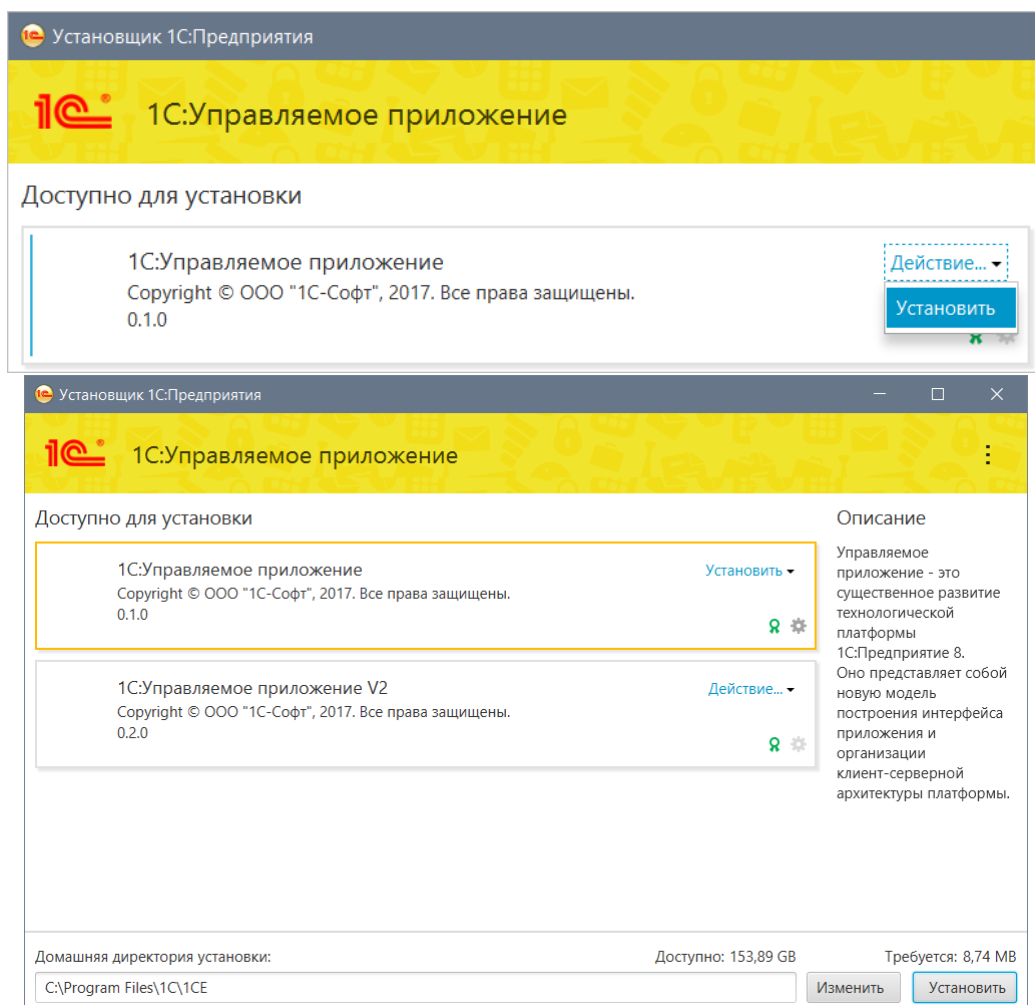
ИСПОЛЬЗОВАНИЕ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА

Графический вариант Установщика ориентирован, прежде всего, на простоту использования и минимальное количество действий, требующихся для установки.

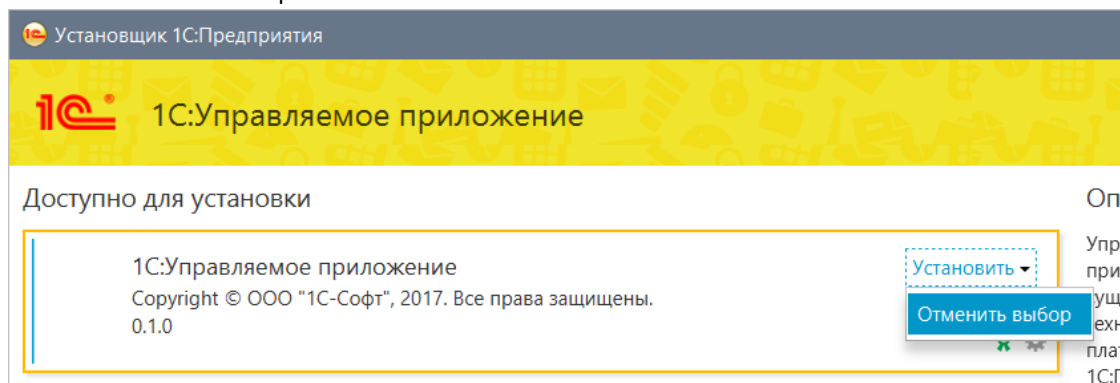
После запуска Установщика на экране отображается список пакетов программного обеспечения, доступных для установки и уже установленных.



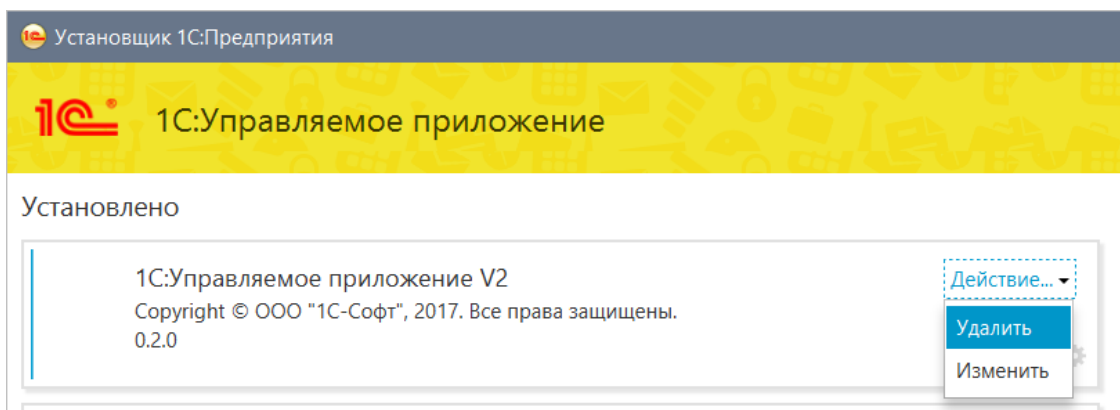
Для выбора желаемого действия необходимо кликнуть на ссылку **Действие** и выбрать желаемое действие. Карточки выбранного программного обеспечения будут выделяться цветом.



Отменить выбранное действие можно через ссылку в той же части карточки, а настроить параметры установки можно с помощью кнопок в нижней части карточки.

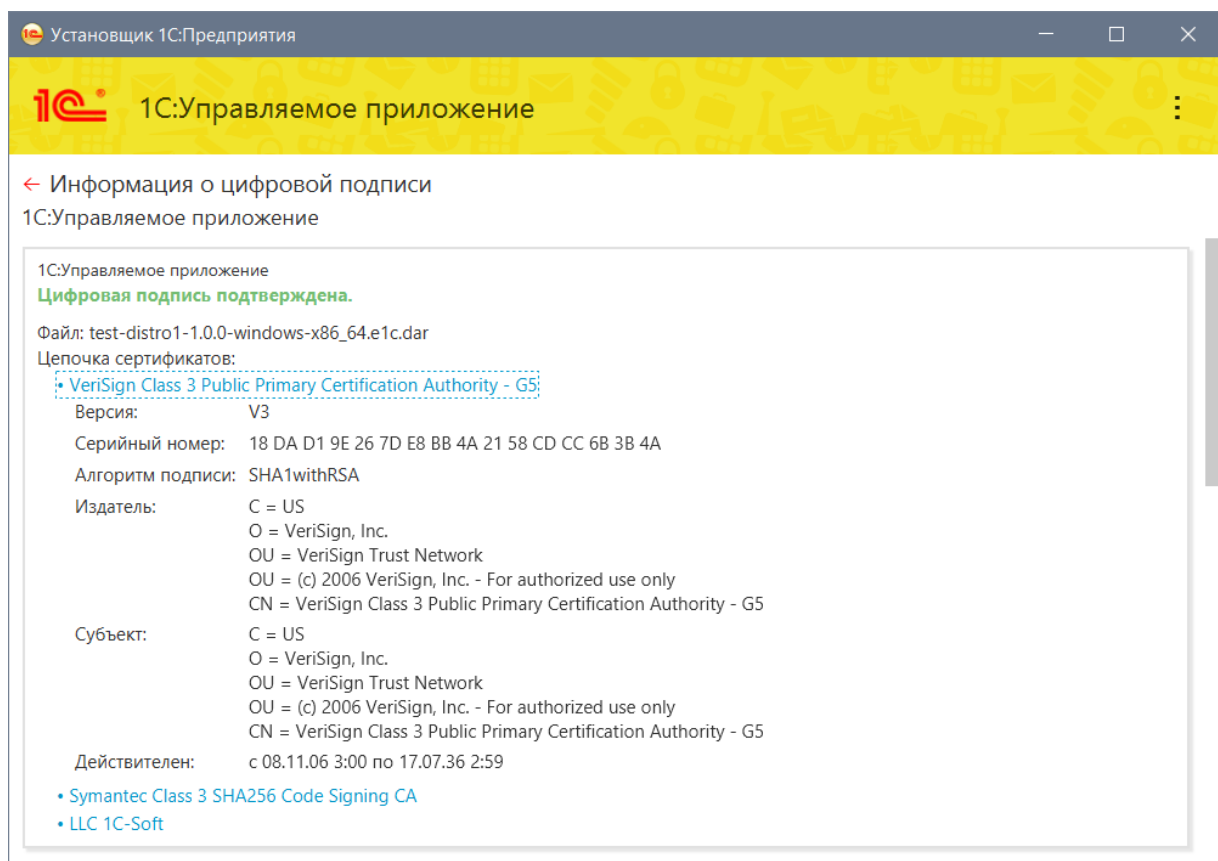


Для удаления или модификации установленного программного обеспечения используется то же меню.



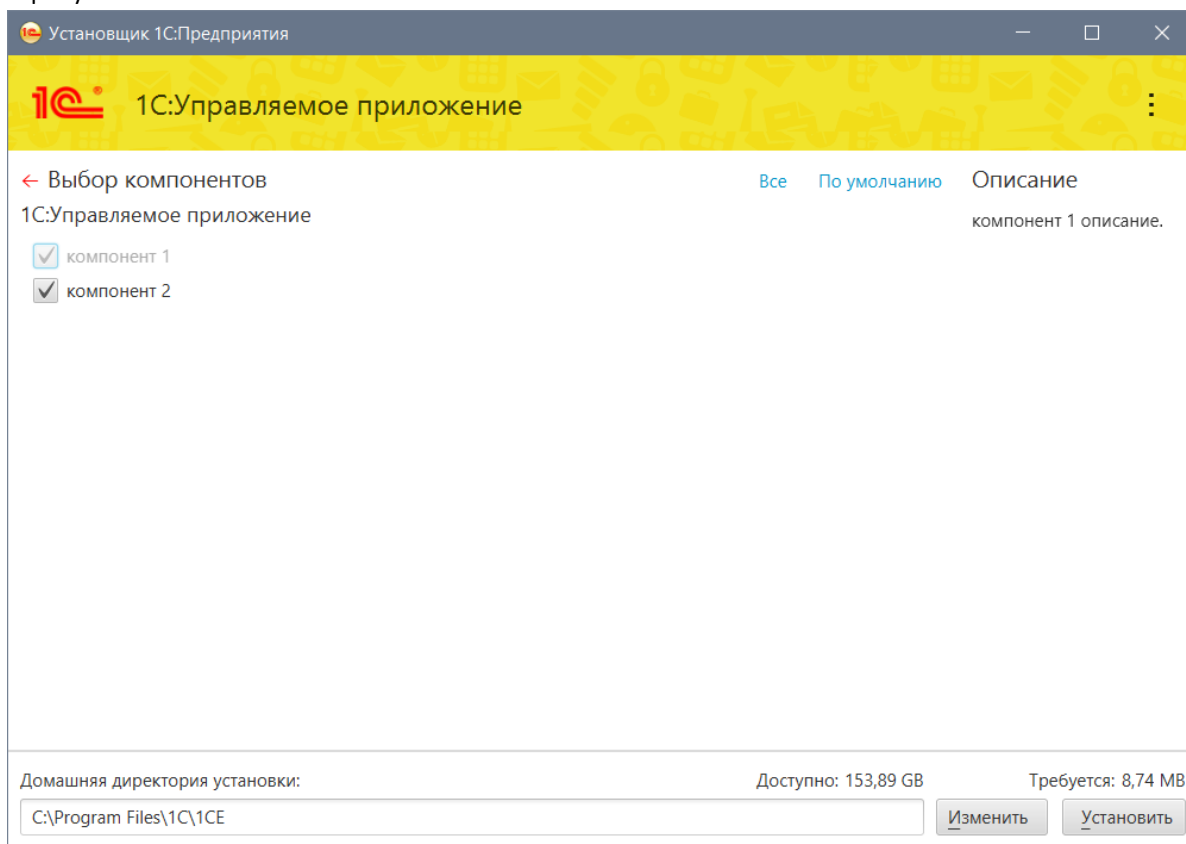
ПРОВЕРКА ЦИФРОВОЙ ПОДПИСИ

Каждый дистрибутив подписан цифровой подписью. Узнать ее статус можно по иконке на карточке для выбора программного обеспечения. Если отображается зеленая "медаль" то все в порядке, и по клику на нее можно просмотреть параметры цифровой подписи.



ВЫБОР КОМПОНЕНТОВ

Каждое программное обеспечение состоит из составных частей, называемых *компонентами*. Если для программного обеспечения возможен выбор компонентов, то на карточке будет кнопка с изображением шестеренки, нажав на которую можно перейти к экрану выбора компонентов. Между компонентами могут быть зависимости и Установщик всегда будет выбирать требуемые зависимости автоматически.



Ссылка **Все** помогает выбрать все компоненты. Ссылка **По умолчанию** приведет к выбору значений по умолчанию. Напротив обязательных для установки компонентов галочка будет всегда выбрана без возможности снятия выбора.

НАСТРОЙКИ ПОЛЬЗОВАТЕЛЕЙ

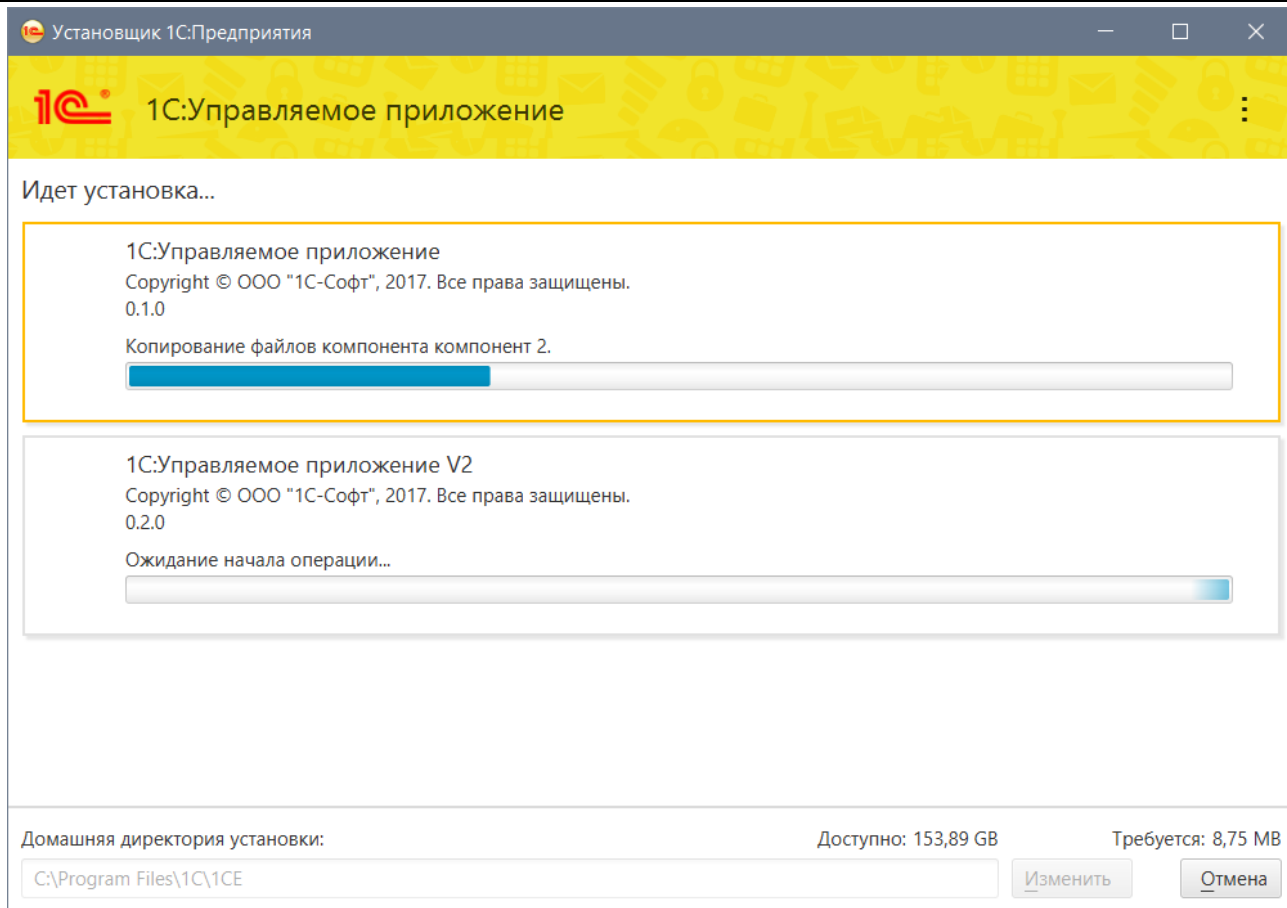
Если для установленного программного обеспечения требуется пользователь, то до начала установки можно настроить параметры пользователя с помощью клика по кнопке с изображением силуэта пользователя.

Это необязательное действие. Если его не совершить, то Установщик создаст специального пользователя или обновит настройки существующего пользователя автоматически. Могут быть затронуты только стандартные для «1С:Предприятия 8» пользователи, например **USR1CE** или **USR1CV8**.

ПРОЦЕСС УСТАНОВКИ

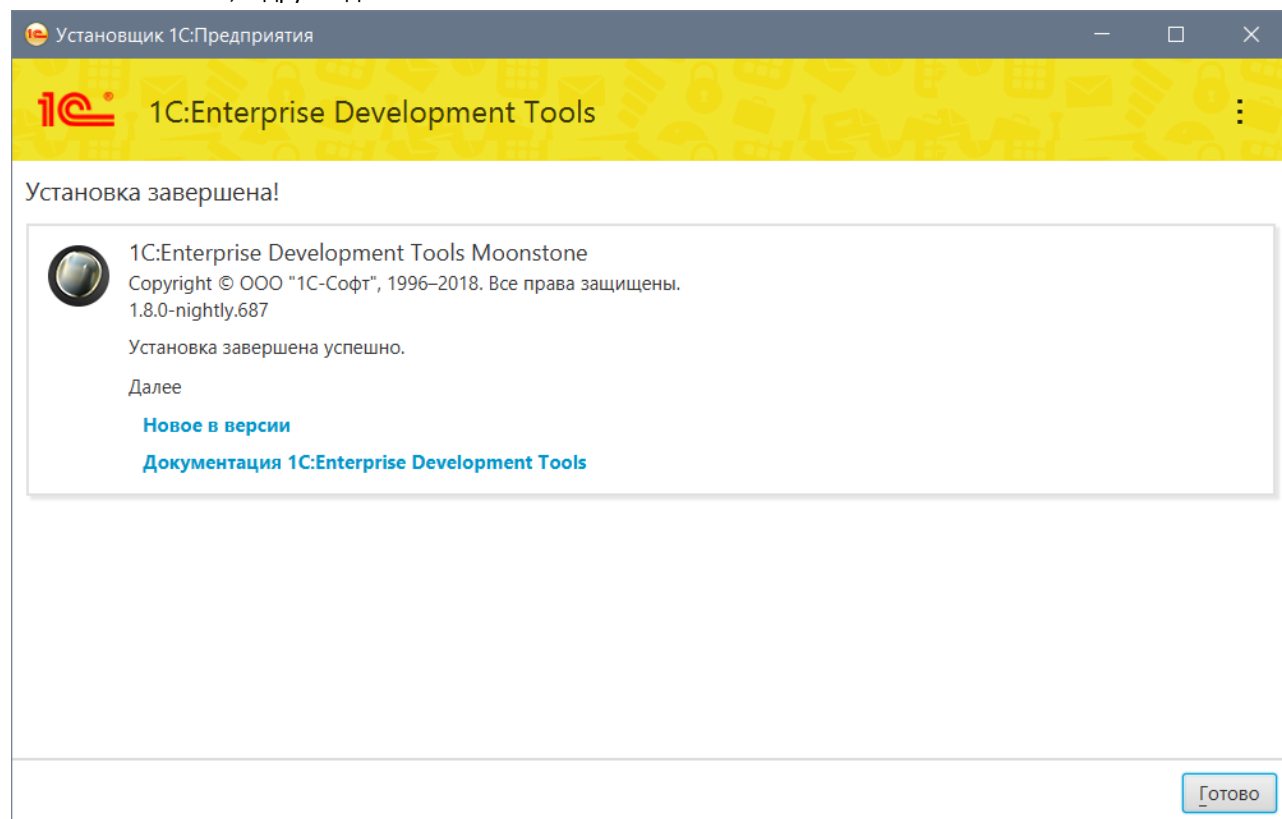
После того как внесены необходимые настройки, процесс установки можно запустить кнопкой в правом нижнем углу окна. В зависимости от выбора программного обеспечения на ней может быть написано **Установить**, **Удалить** или **Применить**. Кнопку можно нажать на любом экране, где она активна, без дополнительных переходов между экранами.

Экран установки выглядит так.



В процессе установки всегда сначала устанавливается новое программное обеспечение, а затем удаляется выбранное для удаления. Цветом выделена карточка программного обеспечения, выбранного на основном экране и для которого в текущий момент происходит установка. На стадии установки нового программного обеспечения процесс установки можно отменить, а на стадии удаления процесс уже прервать невозможно. Отмена происходит по нажатию на соответствующую кнопку или клавишу **ESC**.

После завершения установки на карточках могут появиться ссылки для просмотра README, запуска установленного программного обеспечения, и других действий.

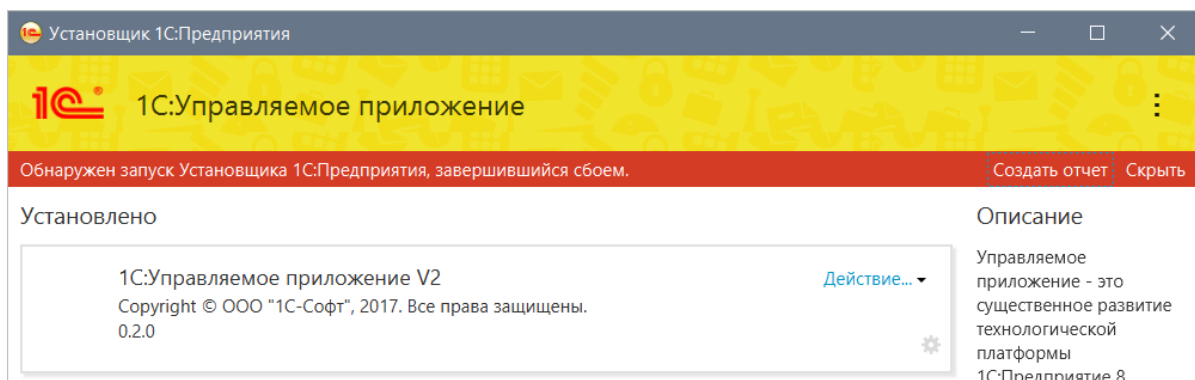


ДИАГНОСТИКА СБОЕВ

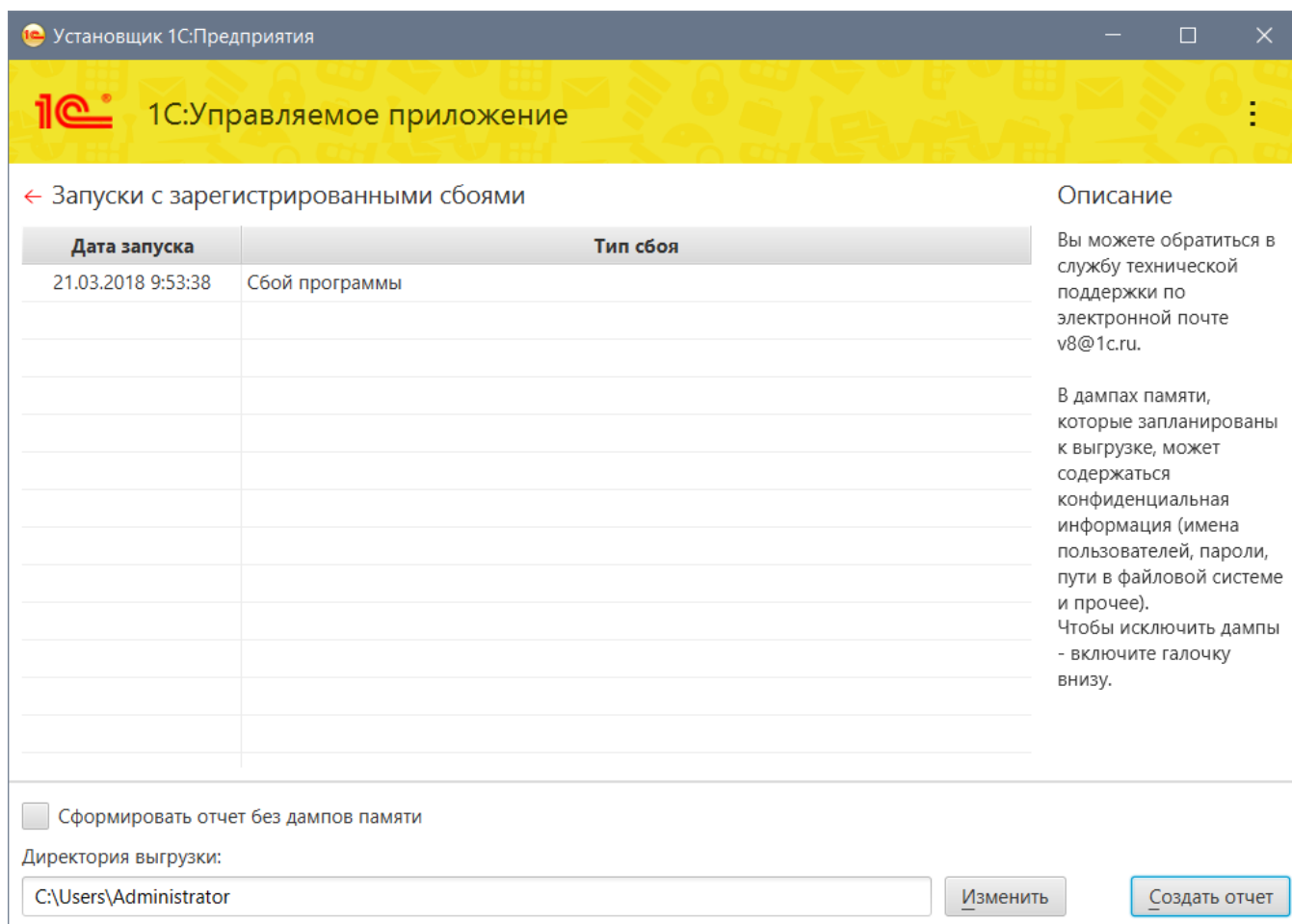
В Установщик встроена возможность сохранить информацию о сбоях в процессе установки и о сбоях самого Установщика, если вдруг такое случится. Вся информация хранится во временной директории системы.

В интерфейсе при этом отображается сообщение, которое либо можно скрыть, либо же можно перейти на экран выгрузки отчетов о сбоях.

Предупреждение выглядит так.



Экран выгрузки отчетов выглядит так.



По нажатию на кнопку **Создать отчет** в указанной директории будет сформирован ZIP-архив с отчетом, который можно отправить в службу технической поддержки.

ПАРАМЕТРЫ ЗАПУСКА

Графический вариант Установщика имеет следующие параметры запуска:

--source

Путь к директории с дистрибутивом. Необязательный параметр. Используется внутренним механизмом перезапуска и может быть использован, например, при необходимости установить какой-либо дистрибутив другой версией Установщика.

--verbose

Определяет уровень журналирования в процессе установки. Может принимать значения `info`, `detailed`, `full`.

Необязательный параметр. По умолчанию используется значение `info`.

Порядок параметров не важен. Значения параметров указываются через пробел, например:

```
lce-installer --verbose full --source c:\path\to\other\distro
```

ИСПОЛЬЗОВАНИЕ КОНСОЛЬНОГО ИНТЕРФЕЙСА

Консольная версия Установщика предназначена в первую очередь для использования в скриптах и интеграции с различными системами управления конфигурацией и не поддерживает интерактивный режим, когда от пользователя ожидаются какие-либо действия.

ПРОСМОТР ВЕРСИИ И СПРАВКИ В КОНСОЛИ

Просмотр версии вызывается командой:

```
lce-installer-cli -version
```

Просмотр полной справки вызывается командой:

```
lce-installer-cli -help
```

Просмотр справки по отдельной команде вызывается командой `lce-installer-cli <команда> --help`. Например:

```
lce-installer-cli install --help
```

ОБЩИЕ ПАРАМЕТРЫ

Консольный вариант Установщика имеет следующие параметры запуска:

--verbose

Определяет уровень журналирования в процессе установки. Может принимать значения `info`, `detailed`, `full`.

Необязательный параметр. По умолчанию используется значение `info`.

Параметры инсталлятора указываются до названия команды и параметров команды:

```
lce-installer-cli <общие параметры> <команда> <параметры команды>
```

Например:

```
lce-installer-cli --verbose full install all
```

КОМАНДА QUERY

Совет: Здесь приведена справка для последней версии Установщика. Справку для используемой версии Установщика можно всегда получить, выполнив в терминале `lce-installer-cli query -help`.

```
lce-installer-cli [<опции>] query [installed] [<параметры>]
```

Ищет продукты или компоненты, удовлетворяющие критериям поиска. Параметры поиска передаются с помощью параметров командной строки. Команда возвращает информацию об установке в определенном YAML формате ([подробнее](#)), который можно передать на вход командам `install` и `uninstall`.

Опции

Для просмотра описания опций необходимо использовать основную справку, введите `lce-installer-cli`.

ПАРАМЕТРЫ

installed

Если задан, то команда ведет поиск среди установленных продуктов. Иначе поиск ведется среди продуктов дистрибутива.

--source

Путь к директории с дистрибутивом. По-умолчанию используется директория, из которой запущен `lce-installer-cli`.

Если задан параметр `installed` то значение параметра `--source` будет проигнорировано.

<products>

Список продуктов для поиска через запятую.

Продукты так же можно задавать, разделяя их параметром `--components`, содержащим более детальные настройки поиска компонентов конкретных продуктов.

Формат для продуктов из дистрибутива:

```
productId[@productVersion]
```

Формат для установленных продуктов:

```
productId[@productVersion] [:productArch]
```

productId

Идентификатор продукта. Обязательный параметр.

productVersion

Версия продукта в формате Semantic Versioning Specification, см. [Приложение 3](#).

Необязательный параметр.

Следует использовать, если более чем один продукт с таким же идентификатором, но другой версией существует в дистрибутиве. Если версия не указана, команда будет искать продукт с любой версией и указанным идентификатором.

--components

Список компонентов продукта для установки через запятую.

Может быть использован только если задан параметр `<products>`.

Должен следовать за одной из секции параметра `<products>`.

Формат:

```
componentId[@componentVersion]
```

Если в предшествующей секции `<products>` указано несколько продуктов, команда будет искать компоненты в каждом из них.

componentId

Идентификатор компонента. Обязательный параметр.

componentVersion

Версия компонента в формате Semantic Versioning Specification, см. <https://semver.org/>.

Необязательный параметр.

Следует использовать, если более чем один компонент с таким же идентификатором, но другой версией существует в продукте в дистрибутиве.

Если версия не указана, команда будет искать продукт с любой архитектурой и указанными идентификатором и версией.

Совет: При указании идентификаторов и версий продуктов и компонентов, а также архитектур продуктов, могут быть использованы специальные символы:

- * — любое количество символов
- ? — ровно один символ
- [...] — один из символов, указанных в квадратных скобках
- [!...] — один символ, не совпадающий ни с одним символом в квадратных скобках

Все значение параметра с такими символами должно быть заключено в двойные кавычки.

ПРИМЕРЫ

```
> lce-installer-cli query
```

Отображает все продукты и компоненты из дистрибутива, который расположен в той же директории что и lce-installer-cli.

```
> lce-installer-cli query --source D:\dist1 prod1@1.0.0:x86
```

Отображает информацию об установке одного продукта с идентификатором **prod1**, версией **1.0.0** и архитектурой **x86** со всеми компонентами из дистрибутива, расположенного в D:\dist1, если этот продукт существует.

```
> lce-installer-cli query installed prod1,prod2
```

Отображает информацию об установке о всех продуктах с идентификаторами **prod1** и **prod2** с их установленными компонентами, если они установлены.

```
> lce-installer-cli query installed "prod*"
```

Отображает информацию об установке о всех продуктах с идентификаторами, удовлетворяющими шаблону **prod***.

```
> lce-installer-cli query prod1 --components c1
```

Отображает информацию об установке о всех продуктах с идентификатором **prod1** с компонентами с идентификатором **c1** из дистрибутива, который расположен в той же директории что и lce-installer-cli.

КОМАНДА INSTALL

Совет: Здесь приведена справка для последней версии Установщика. Справку для используемой версии Установщика можно всегда получить, выполнив в терминале lce-installer-cli install -help.

```
lce-installer-cli [<опции>] install [default|all] [<параметры>]
```

Устанавливает продукты и компоненты. Продукты и компоненты для установки задаются либо в параметрах командной строки, либо в файле.

ОПЦИИ

Для просмотра описания опций необходимо использовать основную справку, введите lce-installer-cli.

ПАРАМЕТРЫ

default

Применяется к продуктам и компонентам.

Если список продуктов не задан, `default` означает все продукты дистрибутива по умолчанию. Если список продуктов задан, то `default` для продуктов игнорируется.

Если список компонентов не задан, то `default` означает все компоненты продуктов по умолчанию. Если список компонентов задан, то `default` игнорируется. Невозможно использовать одновременно с параметром `all`. По-умолчанию используется параметр `default`.

all

Применяется к продуктам и компонентам.

Если список продуктов не задан, `all` означает все продукты дистрибутива. Если список продуктов задан, то `all` для продуктов игнорируется.

Если список компонентов не задан, то `all` означает все компоненты продуктов. Если список компонентов задан, то `all` игнорируется.

Невозможно использовать одновременно с параметром `default`.

По-умолчанию используется параметр `default`.

--source

Путь к директории с дистрибутивом. По-умолчанию используется директория, из которой запущен `lce-installer-cli`.

--products-home

Путь к домашней директории продуктов. Параметр игнорируется, если домашняя директория продуктов уже была задана на этом хосте.

Если не задано, будет использована домашняя директория программных продуктов по-умолчанию.

--overwrite

Если задан то директория установки продуктов/компонентов будет перезаписана. По-умолчанию директория установки не перезаписывается.

--ignore-signature-warnings

Если задан то все проблемы с подписью в дистрибутиве будут проигнорированы. По-умолчанию проблемы с подписью не игнорируются и препятствуют установке.

--ignore-hardware-checks

Если задано то все проверки оборудования будут пропущены. По-умолчанию проверки оборудования включены.

--file

Файл, описывающий продукты для установки в определенном YAML формате ([подробнее](#)).

Не может быть использован вместе с параметром `<products>` одновременно.

Если указано "-", то `lce-installer-cli` будет читать список продуктов для установки из стандартного потока ввода (`stdin`).

Если ни один из них не указан, то команда установит продукты и компоненты соответственно выбранным параметрам `all` или `default`.

<products>

Список продуктов для установки через запятую.

Не может быть использован вместе с параметром `<file>` одновременно.

Если ни один из них не указан, то команда установит продукты и компоненты соответственно выбранным параметрам `all` или `default`.

Продукты так же можно задавать, разделяя их параметром `--components`, содержащим более детальные настройки установки компонентов конкретных продуктов.

Формат:

```
productId[@productVersion]
```

Если продукт не существует в дистрибутиве - команда завершится с ошибкой.

productId

Идентификатор продукта. Обязательный параметр.

productVersion

Версия продукта в формате Semantic Versioning Specification, см . <https://semver.org/>.

Необязательный параметр.

Следует использовать, если более чем один продукт с таким же идентификатором, но другой версией существует в дистрибутиве.

Если версия не указана, но в дистрибутиве существует более одного продукта с таким идентификатором, но разными версиями – команда установит их все.

--components

Список компонентов продукта для установки через запятую.

Может быть использовать только если задан параметр `<products>`.

Должен следовать за одной из секции параметра `<products>`.

Формат:

```
componentId[@componentVersion]
```

Если компонент не существует в продукте в дистрибутиве – команда завершится с ошибкой. Если в предшествующей секции `<products>` указано несколько продуктов, компоненты должны существовать в каждом из них.

componentId

Идентификатор компонента. Обязательный параметр.

componentVersion

Версия компонента в формате Semantic Versioning Specification, см. <https://semver.org/>.

Необязательный параметр.

Следует использовать, если более чем один компонент с таким же идентификатором, но другой версией существует в продукте в дистрибутиве. Если версия не указана, но в продукте в дистрибутиве существует более одного компонента с таким идентификатором, но разными версиями - команда установит их все.

Совет: При указании идентификаторов и версий продуктов и компонентов могут быть использованы специальные символы:

- * — любое количество символов
- ? — ровно один символ
- [...] — один из символов, указанных в квадратных скобках
- [!...] — один символ, не совпадающий ни с одним символом в квадратных скобках

Все значение параметра с такими символами должно быть заключено в двойные кавычки.

ПРИМЕРЫ

```
> lce-installer-cli install
```


Устанавливает продукты по-умолчанию с компонентами по-умолчанию из дистрибутива, который расположен в той же директории что и `lce-installer-cli`.

```
> lce-installer-cli install prod1@1.0.0
```

Устанавливает один продукт с идентификатором **prod1**, версией **1.0.0** и архитектурой и ОС такими же как дистрибутив из которого был запущен `lce-installer-cli` и с компонентами по-умолчанию, если этот продукт существует в дистрибутиве.

```
> lce-installer-cli install all prod1,prod2
```

Устанавливает продукты с идентификаторами **prod1** и **prod2** со всеми компонентами, если эти продукты существуют в дистрибутиве. Если в дистрибутиве несколько продуктов с идентификатором **prod1** и/или **prod2**, то установятся все версии этих продуктов.

```
> lce-installer-cli install all "prod*"
```

Устанавливает все продукты из дистрибутива, идентификаторы которых удовлетворяют шаблону **prod***.

```
> lce-installer-cli install --file D:\uninst.yml --source C:\dist\dist1
```

Устанавливает продукты, описанные в файле `D:\uninst.yml` и из дистрибутива расположенного в `C:\distros\distro1`.

```
> lce-installer-cli install prod1 --components c1 prod2 --components c2
```

Устанавливает продукт с идентификатором **prod1** и с компонентом **c1**, а так же продукт **prod2** с компонентом **c2**, если они существуют в дистрибутиве в соответствующих продуктах.

```
> lce-installer-cli --verbose full install --ignore-signature-warnings
```

Устанавливает продукты по-умолчанию с продуктами по-умолчанию из дистрибутива, который расположен в той же директории что и `lce-installer-cli` и игнорирует все проблемы с подписью, существующие в этом дистрибутиве. Так же устанавливает уровень логгирования в значение **full**.

```
> cat /tmp/inst.yml | lce-installer-cli install --source /dist1 --file -
```

Устанавливает продукты, описанные в файле `/tmp/inst.yml` и из дистрибутива расположенного в `/dist1`.

КОМАНДА UNINSTALL

Совет: Здесь приведена справка для последней версии Установщика. Справку для используемой версии Установщика можно всегда получить, выполнив в терминале `lce-installer-cli uninstall -help`.

```
lce-installer-cli [<options>] uninstall <parameters>
```

Удаляет продукты и компоненты с диска. Продукты и компоненты для удаления задаются либо в параметрах командной строки, либо в файле.

ОПЦИИ

Для просмотра описания опций необходимо использовать основную справку, введите `lce-installer-cli`.

ПАРАМЕТРЫ

--file

Файл, описывающий продукты для удаления в определенном YAML формате ([подробнее](#)). Если указано "-", то `1ce-installer-cli` будет читать список продуктов для удаления из стандартного потока ввода (`stdin`). Не может быть использован вместе с параметром `<products>` одновременно. Один из этих параметров должен быть указан.

<products>

Список продуктов для удаления через запятую. Не может быть использован вместе с параметром `<file>` одновременно. Один из этих параметров должен быть указан.

Формат:

```
productId[@productVersion] [:productArch]
```

Если продукт не существует - команда завершится с ошибкой.

productId

Идентификатор продукта. Обязательный параметр.

productVersion

Версия продукта в формате Semantic Versioning Specification, см. <https://semver.org/>.

Необязательный параметр.

Следует использовать, если существует более чем один продукт с таким же идентификатором, но другой версией.

Если версия не указана, но существует более одного продукта с таким идентификатором, но разными версиями - команда удалит их все.

productArch

Архитектура продукта. Возможные значения: `x86`, `x86_64`.

Необязательный параметр.

Следует использовать, если существует более чем один продукт с таким же идентификатором и версией, но другой архитектурой.

Если архитектура не указана, но существует более одного продукта с таким идентификатором и версией, но разными архитектурами - команда удалит их все.

--components

Список компонентов продукта для удаления через запятую.

Может быть использовать только если задан параметр `<products>`.

Должен следовать за одной из секции параметра `<products>`.

Формат:

```
componentId[@componentVersion]
```

Если компонент не существует в продукте в дистрибутиве – команда завершится с ошибкой. Если в предшествующей секции `<products>` указано несколько продуктов, компоненты должны существовать в каждом из них.

Невозможно удалить обязательный компонент, если только не удаляется весь продукт.

Невозможно удалить компонент, на который имеет зависимость какой-либо другой компонент из этого продукта, который при этом не удаляется.

componentId

Идентификатор компонента. Обязательный параметр.

componentVersion

Версия компонента в формате Semantic Versioning Specification, см. <https://semver.org/>.

Необязательный параметр.

Следует использовать, если более чем один компонент с таким же идентификатором, но другой версией существует в продукте.

Если версия не указана, но в продукте существует более одного компонента с таким идентификатором, но разными версиями - команда удалит их все.

Совет: При указании идентификаторов и версий продуктов и компонентов могут быть использованы специальные символы:

- * — любое количество символов
- ? — ровно один символ
- [...] — один из символов, указанных в квадратных скобках
- [...] — один символ, не совпадающий ни с одним символом в квадратных скобках

Все значение параметра с такими символами должно быть заключено в двойные кавычки.

ПРИМЕРЫ

```
> lce-installer-cli uninstall prod1@1.0.0:x86
```

Удаляет один продукт с идентификатором **prod1**, версией **1.0.0** и архитектурой **x86**, если этот продукт существует.

```
> lce-installer-cli uninstall prod1,prod2
```

Удаляет оба продукта с идентификаторами **prod1** and **prod2**, если они существуют. Если установлено несколько продуктов с идентификатором **prod1** и/или **prod2**, то удаляются все продукты всех версий и архитектур с идентификатором **prod1** и/или **prod2**.

```
> lce-installer-cli uninstall "prod*"
```

Удаляет продукты всех версий и архитектур, идентификатор которых соответствует шаблону **prod***.

```
> lce-installer-cli uninstall --file D:\uninstall.yml
```

Удаляет продукты, описанные в файле D:\uninstall.yml.

```
> cat /tmp/uninstall.yml | lce-installer-cli uninstall --file -
```

Удаляет продукты, описанные в файле /tmp/uninstall.yml.

ПОДСИСТЕМА SUPPORT

Совет: Здесь приведена справка для последней версии Установщика. Справку для используемой версии Установщика можно всегда получить, выполнив в терминале `lce-installer-cli support --help` и `lce-installer-cli support failures --help`.

Подсистема **support** содержит несколько команд:

lce-installer-cli support clean

Удаляет информацию о предыдущих запусках Установщика, включая информацию о запусках, завершившихся сбоем.

lce-installer-cli support failures list

Отображает в консоли информацию о прошлых сбоях Установщика.

1ce-installer-cli support failures clean

Очистка всей информации о прошлых сбоях Установщика.

1ce-installer-cli support failures suppress

Подавляет уведомление о прошлых запусках, завершившихся сбоем. При этом, информация о сбое не удаляется и может быть выгружена командой export.

1ce-installer-cli support failures export [--directory < директория_выгрузки>] [--exclude-memory-dumps]

Выгружает информацию о прошлых сбоях в указанную директорию.

--directory

Директория, в которую будет выгружен ZIP- архив с информацией о предыдущих сбоях Установщика. Если параметр не указан, то информация выгружается в текущую директорию.

--exclude-memory-dumps

Если `--exclude-memory-dumps` указан, дампы памяти не будут экспортированы. По умолчанию, дампы памяти экспортируются. Дампы памяти могут содержать конфиденциальную информацию, включая логины, пароли и т. п.

ЗАПУСК И УДАЛЕНИЕ УСТАНОВЩИКА ПОСЛЕ УСТАНОВКИ**ЗАПУСК УСТАНОВЛЕННОГО УСТАНОВЩИКА**

При установке любого программного обеспечения на диск компьютера так же копируется Установщик.

Для удобства запуска Установщика ярлык для него добавлен в стартовое меню Windows и в Launchpad на macOS (в группе **1С**). Для Linux в настоящее время ярлык Установщика не создается, и Установщик необходимо будет запускать из командной строки.

Запустить Установщик из консоли или проводника можно как и при первоначальной установке. Файлы Установщика расположены в директории `components/1c-enterprise-installer-<версия>-<архитектура>` внутри домашней директории программных продуктов.

УДАЛЕНИЕ УСТАНОВЛЕННОГО УСТАНОВЩИКА

В настоящий момент удаление возможно только вручную. Мы понимаем, что это некоторый недостаток для тех, кто хочет только попробовать новое программное обеспечение, и планируем устранить его в ближайших версиях.

Перед удалением Установщика обязательно необходимо удалить все установленное программное обеспечение с его помощью.

Далее необходимо удалить (все пути приведены по умолчанию):

Домашняя директория программных продуктов

- Windows — %ProgramFiles%\1C\1CE
- Linux — /opt/1C/1CE
- macOS — /Applications/1C/1CE

Реестр программных продуктов

- файлы `inventory.loc` и `inventory.write.lock`
- директорию `inventory`
- расположение файлов:
 - Windows — %ProgramData%\1C\1CE
 - Linux — /etc/1C/1CE
 - macOS — /Library/Application Support/1C/1CE

Журнальные файлы, если ОС не настроена на их автоматическое удаление

- Windows — все директории по маске `%TEMP%\1ce-installer*`
- Linux и macOS — все директории `/tmp/1ce-installer*`

ОСОБЕННОСТИ ПОДДЕРЖКИ РАЗЛИЧНЫХ ОПЕРАЦИОННЫХ СИСТЕМ

ОСОБЕННОСТИ ПОДДЕРЖКИ MACOS

Для macOS рекомендуется не изменять домашнюю директорию программных продуктов на такую, которая не является поддиректорией в /Applications. Установщик будет корректно работать с любой директорией, однако macOS в таком случае не будет автоматически добавлять установленные приложения на экран Launchpad и их придется запускать вручную. Если есть необходимость изменить домашнюю директорию программных продуктов, то возможным обходным решением является ручное создание символических ссылок на приложения в /Applications.

ОСОБЕННОСТИ ПОДДЕРЖКИ LINUX

Для Linux создано огромное количество оболочек (desktop environment) несовместимых между собой. Это сказывается на механизме создания ярлыков для установленного программного обеспечения. Поэтому при разработке мы ориентировались на спецификацию Desktop Menu Specification от FreeDesktop (<https://specifications.freedesktop.org/menu-spec/latest/>) и проверяли работу на Gnome 3 и KDE 5.

Также в Linux возможны проблемы с отображением иконки в панели задач. Это связано с ограниченной поддержкой данной возможности в JavaFX.

ПРИЛОЖЕНИЕ 1. ФОРМАТ ФАЙЛА INVENTORY.LOC

Файл `inventory.loc` указывает на расположение реестра программных продуктов. Формат файла — YAML. Кодировка содержимого файла — UTF-8. Расположение файла:

- Windows — %ProgramData%\1C\1CE\inventory.loc
- Linux — /etc/1C/1CE/inventory.loc
- macOS — /Library/Application Support/1C/1CE/inventory.loc

Пример файла `inventory.loc`:

```
inventory-location

inventory-root-path: C:\ProgramData\1C\1CE\Inventory

inventory-version: 42
```

Если необходимо перенести реестр программных продуктов, то после переноса следует отредактировать данный файл и указать новое расположение.

ПРИЛОЖЕНИЕ 2. ФОРМАТ YAML-ФАЙЛА ДЛЯ УСТАНОВКИ И УДАЛЕНИЯ С ПОМОЩЬЮ КОНСОЛЬНОЙ ВЕРСИИ

Настройки установки и удаления для консольных команд Установщика можно сохранять в файлы в формате YAML.

В файле можно описать несколько YAML-документов, разделенных тремя знаками (минус), каждый из которых описывает параметры одного продукта. В зависимости от команды (`install` или `uninstall`) параметры из файла будут трактоваться соответственно для создания или для удаления.

Пример файла:

```
id: product1

version: 1.2.17+112

arch: x86_64

components:

  - aaa@0.10.0
```

```
- bbb@1.0.0

user:

  username: testuser

  password: testpassword

  description: Test user

---

id: product2

version: 2.4.9+31

arch: x86_64

components:

  - ccc@0.10.0

  - bbb@1.0.0
```

Описание атрибутов файла:

id

Идентификатор продукта (программного обеспечения).
Обязательный параметр.

version

Версия продукта в формате semver.

Необязательный параметр. Если в дистрибутиве имеется несколько продуктов с одинаковым `id` и разной версией и параметр не указан, то будут установлены все продукты. Аналогично при удалении — если не указана версия, то будут удалены все продукты с указанным `id`.

arch

Архитектура продукта. Возможные значения `x86` и `x86_64`.

Необязательный параметр. При установке вычисляется на основании дистрибутива. Если не указана при удалении, то будут удалены продукты для обеих архитектур.

components

Список компонентов.

Не обязательный параметр. Если список не указан то при установке будут установлен набор компонентов по умолчанию, а при удалении будут удалены все установленные компоненты для данного продукта. Если список указан, то будут установлены или удалены только указанные компоненты.

Элемент списка `components` указывает на выбор компонента для установки.

Строка в виде ID@version, где **ID** - идентификатор компонента, а **version** — его версия в формате semver. Например `1c-enterprise-ring@0.10.0`

user

Описание создаваемого пользователя ОС.

Используется только при установке и только если для устанавливаемого продукта можно создать пользователя. В остальных случаях игнорируется.

Узнать, требуется ли продукту пользователь, можно выполнив команду `lce-installer-cli query` в директории дистрибутива.

Атрибуты пользователя (**user**):

username

Логин пользователя ОС. Используется только при установке.

password

Пароль пользователя ОС. Используется только при установке. Если пользователь существует, то его пароль изменен не будет.

description

Описание пользователя. Необязательный параметр. Не должно содержать спец. символов.

primary-group

Группа пользователя. Используется только при установке на ОС Linux. Если пользователь существует, то его группа изменена не будет.

additional-groups

Дополнительные группы пользователя. Список строк. Используется только при установке на ОС Linux. Если пользователь существует, то он будет включен во все перечисленные здесь группы.

ПРИЛОЖЕНИЕ 3. СПЕЦИФИКАЦИЯ СЕМАНТИЧЕСКОГО ВЕРСИОНИРОВАНИЯ ([SEMVER](#))

Слова «ДОЛЖЕН» (MUST), «НЕ ДОЛЖЕН» (MUST NOT), «ОБЯЗАТЕЛЬНО» (REQUIRED), «СЛЕДУЕТ» (SHOULD), «НЕ СЛЕДУЕТ» (SHOULD NOT), «РЕКОМЕНДОВАННЫЙ» (RECOMMENDED), «МОЖЕТ» (MAY) и «НЕОБЯЗАТЕЛЬНЫЙ» (OPTIONAL) в этом документе должны быть интерпретированы в соответствии с [RFC 2119](#).

1. ПО, использующее Семантическое Версионирование, должно объявить публичный API. Этот API может быть объявлен самим кодом или существовать строго в документации. Как бы ни было это сделано, он должен быть точным и исчерпывающим.
2. Обычный номер версии ДОЛЖЕН иметь формат X.Y.Z, где X, Y и Z — неотрицательные целые числа и НЕ ДОЛЖНЫ начинаться с нуля. X — мажорная версия, Y — минорная версия и Z — патч-версия. Каждый элемент ДОЛЖЕН увеличиваться численно. Например: 1.9.0 -> 1.10.0 -> 1.11.0.
3. После релиза новой версии пакета содержание этой версии НЕ ДОЛЖНО быть модифицировано. Любые изменения ДОЛЖНЫ быть выпущены как новая версия.
4. Мажорная версия ноль (0.y.z) предназначена для начальной разработки. Всё может измениться в любой момент. Публичный API не должен рассматриваться как стабильный.
5. Версия 1.0.0 определяет публичный API. После этого релиза номера версий увеличиваются в зависимости от того, как изменяется публичный API.
6. Патч-версия Z (x.y.z | x > 0) ДОЛЖНА быть увеличена только если содержит обратно совместимые баг-фиксы. Определение баг-фикс означает внутренние изменения, которые исправляют некорректное поведение.
7. Минорная версия (x.Y.z | x > 0) ДОЛЖНА быть увеличена, если в публичном API представлена новая обратно совместимая функциональность. Версия ДОЛЖНА быть увеличена, если какая-либо функциональность публичного API помечена как устаревшая (deprecated). Версия МОЖЕТ быть увеличена в случае реализации новой функциональности или существенного усовершенствования в приватном коде. Версия МОЖЕТ включать в себя изменения, характерные для патчей. Патч-версия ДОЛЖНА быть обнулена, когда увеличивается минорная версия.
8. Мажорная версия X (X.y.z | X > 0) ДОЛЖНА быть увеличена, если в публичном API представлены какие-либо обратно несовместимые изменения. Она МОЖЕТ включать в себя изменения, характерные для уровня минорных версий и патчей. Когда увеличивается мажорная версия, минорная и патч-версия ДОЛЖНЫ быть обнулены.
9. Предрелизная версия МОЖЕТ быть обозначена добавлением дефиса и серией разделённых точкой идентификаторов, следующих сразу за патч-версией. Идентификаторы ДОЛЖНЫ содержать только ASCII буквенно-цифровые символы и дефис [0-9A-Za-z-]. Идентификаторы НЕ ДОЛЖНЫ быть пустыми. Числовые идентификаторы НЕ ДОЛЖНЫ начинаться с нуля. Предрелизные версии имеют более низкий приоритет, чем соответствующая релизная версия. Предрелизная версия указывает на то, что эта версия не стабильна и может не удовлетворять требованиям совместимости, обозначенными соответствующей нормальной версией. Примеры: 1.0.0-alpha, 1.0.0-alpha.1, 1.0.0-0.3.7, 1.0.0-x.7.z.92.

10. Сборочные метаданные МОГУТ быть обозначены добавлением знака плюс и ряда разделённых точкой идентификаторов, следующих сразу за патчем или предрелизной версией. Идентификаторы ДОЛЖНЫ содержать только ASCII буквенно-цифровые символы и дефис [0-9A-Za-z-]. Идентификаторы НЕ ДОЛЖНЫ быть пустыми. Сборочные метаданные СЛЕДУЕТ игнорировать, когда определяется старшинство версий. Поэтому два пакета с одинаковой версией, но разными сборочными метаданными, рассматриваются как одна и та же версия. Примеры: 1.0.0-alpha+001, 1.0.0+20130313144700, 1.0.0-beta+exp.sha.5114f85.
11. Приоритет определяет, как версии соотносятся друг с другом, когда упорядочиваются. Приоритет версий ДОЛЖЕН рассчитываться путём разделения номеров версий на мажорную, минорную, патч и предрелизные идентификаторы. Именно в такой последовательности (сборочные метаданные не фигурируют в расчёте). Приоритет определяется по первому отличию при сравнении каждого из этих идентификаторов слева направо: Мажорная, минорная и патч-версия всегда сравниваются численно. Пример: 1.0.0 < 2.0.0 < 2.1.0 < 2.1.1. Когда мажорная, минорная и патч-версия равны, предрелизная версия имеет более низкий приоритет, чем нормальная версия. Пример: 1.0.0-alpha < 1.0.0. Приоритет двух предрелизных версий с одинаковыми мажорной, минорной и патч-версией ДОЛЖНЫ быть определены сравнением каждого разделённого точкой идентификатора слева направо до тех пор, пока различие не будет найдено следующим образом: идентификаторы, состоящие только из цифр, сравниваются численно; буквенные идентификаторы или дефисы сравниваются лексически в ASCII-порядке. Численные идентификаторы всегда имеют низший приоритет, чем символьные. Большой набор предрелизных символов имеет больший приоритет, чем меньший набор, если сравниваемые идентификаторы равны. Пример: 1.0.0-alpha < 1.0.0-alpha.1 < 1.0.0-alpha.beta < 1.0.0-beta < 1.0.0-beta.2 < 1.0.0-beta.11 < 1.0.0-rc.1 < 1.0.0.