

# Приложение 4. Вспомогательные утилиты

## 4.1. Утилита тестирования chdbfl

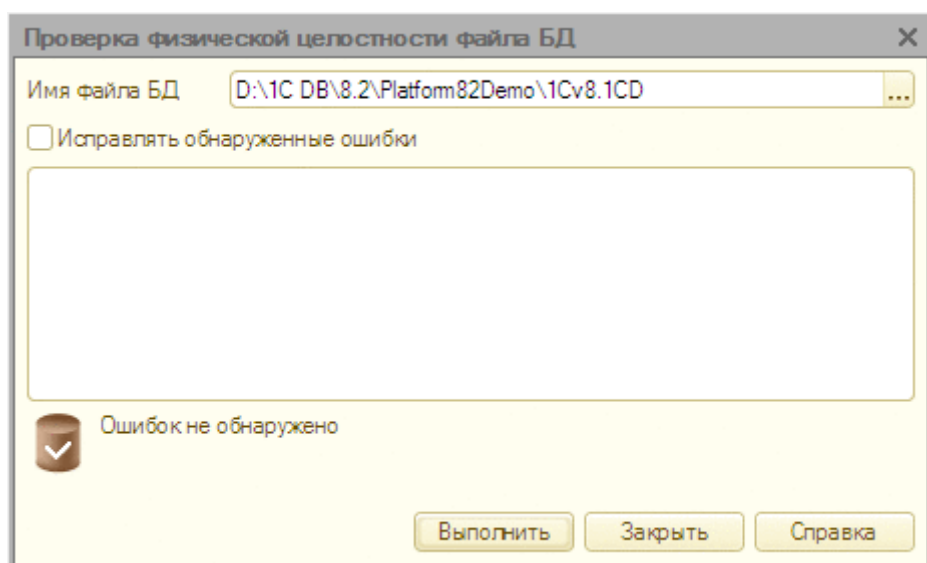
Утилита предназначена для автономной проверки и исправления файлов базы данных.

**ВНИМАНИЕ! Перед использованием данной утилиты следует обязательно сделать резервную копию файла базы данных.**

Данная утилита предназначена для работы только с файловой базой данных. Утилита доступна только в 32-разрядном варианте. Файловая база данных используется:

- для хранения информационной базы в файловом варианте;
- хранения хранилища конфигурации.

Для запуска утилиты в каталоге установки системы «1С:Предприятие» нужно запустить приложение [chdbfl](#). На экран выводится окно (см. [рис.119](#)).



**Рис. 119. Утилита тестирования и исправления информационных баз**

В поле [Имя файла БД](#) указать или выбрать имя файла информационной базы.

Установить флажок [Исправлять обнаруженные ошибки](#), если требуется исправлять обнаруженные при проверке ошибки. Также этот флажок следует установить, если необходимо выполнить оптимизацию размещения служебной информации, ускоряющей открытие информационной базы (см. [здесь](#)).

Для начала работы утилиты нужно нажать кнопку [Выполнить](#). К этому моменту выбранная информационная база не должна быть открыта конфигуратором или в режиме 1С:Предприятие.

Сообщения о найденных ошибках выводятся в текстовое поле.

Ниже текстового поля выводятся сообщения о результатах работы утилиты.

Также эта утилита может использоваться и для проверки хранилища конфигурации.

Рекомендуется использовать следующую последовательность шагов при использовании утилиты:

- **Обязательно создать резервную копию файла базы данных.**
- Выполнить тестирование без установленного флажка [Исправлять обнаруженные ошибки](#).
- Если при тестировании не выявлено проблем, то можно выполнить тестирование с установленным флажком [Исправлять обнаруженные ошибки](#) - при этом будет выполнена операция реиндексации информационной базы или информационной базы хранилища. Такую операцию рекомендуется выполнять регулярно.

- Если при тестировании обнаружены проблемы, то попытаться выполнить исправление информационной базы. Если в процессе работы утилита сообщила о том, что какие-либо данные потеряны, то работать с такой информационной базой не рекомендуется. Из получившейся информационной базы можно:
  - получить сохранившиеся данные и конфигурацию для создания на их основе новой информационной базы;
  - получить последнюю версию конфигурации, на основании которой создать другое хранилище.

## 4.2. Утилита контроля целостности ci

### 4.2.1. Общая информация

Утилита контроля целостности (**ci**) предназначена для контроля состояния объектов файловой системы и базы данных, используемых при работе «1С:Предприятие», и обнаружения ситуации изменения этих объектов. Для определения факта неизменности объекта используется сравнение хеш-сумм контролируемых объектов (**объектов контроля**), которые вычислены по алгоритму SHA-1 (алгоритм криптографического хеширования). Процесс проверки состоит из формирования эталонных значений хеш-сумм и последующей регулярной проверки.

**ПРИМЕЧАНИЕ.** Не поддерживается работа под управлением ОС macOS.

Утилита работает со следующими объектами контроля:

- Файлы, расположенные в файловой системе;
- Некоторые таблицы базы данных системы «1С:Предприятие».

Списки объектов контроля для работы утилиты задаются с помощью **шаблона источника информации**. Шаблон источника информации описывается **универсальным описателем источника информации** (см. [здесь](#)). Такие шаблоны могут указываться в командной строке запуска утилиты непосредственно или как содержимое файла, указанного в качестве значения параметра **in** (см. [здесь](#)). Такой файл будет называться **списком шаблонов** (см. [здесь](#)).

Перечень объектов контроля и соответствующих хеш-сумм (такая пара называется **эталон**) хранится в специальной **базе эталонов**, который формируется утилитой при запуске в режиме формирования базы эталонов. При запуске утилиты в режиме проверки выполняется вычисление хеш-сумм и сверка вычисленных хеш-сумм с ранее сформированной базой эталонов. В результате формируется отчет о работе в виде файла.

Утилита при запуске (или в каких-либо других режимах) автоматически не контролирует собственную целостность. Ограничение доступа к утилите, базе эталонов и результатам проверок следует осуществлять средствами операционной системы.

### 4.2.2. Список шаблонов источников информации

При необходимости указания для контроля одновременно нескольких объектов контроля (или набора объектов) можно использовать специальный файл - список шаблонов источников информации. Каждая строка в этом файле содержит шаблон источника информации. Количество строк в файле не ограничено. Подробное описание формата шаблона источника информации см. [здесь](#). Комментарий начинается с символа "#". Строка комментария, а также пустые строки и строки, состоящие из одних пробелов, игнорируется при формировании списка объектов контроля.

Файл списка шаблонов должен быть только в кодировке UTF-8.

*Примеры универсальных описателей источников информации:*

[Копировать в буфер обмена](#)

```
# объект контроля - файл /tmp/vokas/spru.cvs
file:///tmp/vokas/spru.cvs
# объект контроля - все файлы в каталоге c:\Program Files\1cv8\8.3.4.408\bin
rdir://c:\Program Files\1cv8\8.3.4.408\bin
# объект контроля - все файлы .rc и .cfg в каталоге /home/user/.kde и подчиненных
каталогах
rdir:///home/user/.kde?*.*rc,*.*cfg
```

```
# объект контроля - все файлы, начинающиеся с "V8" файлы в каталоге /tmp, без подкаталогов
ndir:///tmp?V8*.txt
# объект контроля - все поддерживаемые таблицы в базе данных, расположенной на MS SQL Server
mssql://user:password@server/instance/dbname
# объект контроля - таблицы users и config на сервер PostgreSQL
postgres://user:password@server:123/dbname?users,config
# объект контроля - таблицы users в файловой базе данных
dbe://c:\DB\checked_db?users
```

### 4.2.3. Универсальный описатель источника информации

Для описания объекта контроля используется специальный формат - универсальный описатель источника информации. В общем виде универсальный описатель источника информации выглядит следующим образом:

[proto://\[user:\[password\]@\]\[server\[:port\]\]\[/path/resource\]\[?mask\]](#)

Подробнее рассмотрим каждую составную часть описателя:

- **proto** - описание типа объекта контроля. В качестве способа работы могут выступать:
  - **file** - отдельный файл в файловой системе;
  - **ndir** - файлы, расположенные в каталоге (без обхода подкаталогов);
  - **rdir** - файлы, расположенные в каталоге и всех вложенных каталогах;
  - **mssql** - база данных, расположенная в СУБД Microsoft SQL Server;
  - **ora** - база данных, расположенная в СУБД Oracle Database;
  - **postgres** - база данных, расположенная в СУБД PostgreSQL;
  - **db2** - база данных, расположенная в СУБД IBM Db2;
  - **dbe** - файловый вариант базы данных «1С:Предприятия».
- **user:password@** - описывает имя пользователя (**user**) и пароль (**password**), которые требуются для доступа к объекту контроля. Если выполняется контроль таблиц базы данных, то в качестве пользователя указывается пользователь СУБД, а не информационной базы «1С:Предприятия». Рекомендуется использовать параметры того пользователя, от имени которого создавалась база данных «1С:Предприятия». Символ «@» является обязательным в том случае, если указывается имя пользователя и пароль.
- **server:port** - имя компьютера (и порт доступа), на котором запущена СУБД, обслуживающая базу данных «1С:Предприятия»;
- **/path/resource** - полный путь к файлу или каталогу в нотации используемой операционной системы. В ОС Windows должен начинаться с имени диска, в ОС Linux - с обозначения корневой файловой системы (/). Если используется подключение к клиент-серверному варианту базы данных, то в качестве пути выступает имя базы данных в терминах используемой СУБД, если СУБД не поддерживает организацию экземпляров сервера, и комбинация из имени экземпляра и имени базы данных для СУБД, которые такую организацию поддерживают. Если в качестве пути к ресурсу указывается имя базы данных с указанием экземпляра СУБД, то доступ к СУБД осуществляется с использованием порта доступа по умолчанию и указание номера порта в описании имени сервера, на котором функционирует СУБД, не поддерживается.
- **?mask** - перечень параметров описателя источника информации, разделенных запятыми. Зависит от протокола. В том случае, если в качестве объекта контроля выступают файлы, описанные по протоколам **ndir** или **rdir**, то в качестве параметров могут выступать маски файлов. Если объектом контроля выступает таблица базы данных «1С:Предприятия», то в качестве параметров могут выступать имена таблиц:
  - **config** - конфигурация базы данных;
  - **users** - таблица пользователей конфигурации;

Обе этих таблицы являются виртуальными и содержат в себе информацию из нескольких таблиц базы данных «1С:Предприятия». Если в универсальном описателе информации присутствует ссылка на одну из приведенных выше таблиц, то хеш-сумма считается для всех данных, входящих в состав таблицы (физические таблицы, полностью или нет). Возможность контролировать фрагмент таблицы не предоставляется.

**ПРИМЕЧАНИЕ.** При формировании универсального описателя источника информации следует избегать использования следующих символов: @ (кроме случая разделения имени пользователя/пароля и имени компьютера, на котором функционирует СУБД), /, \, :, ? (кроме случая указания в масках файлов, но не ранее), ~.

#### 4.2.4. Файл базы эталонов

Хеш-суммы объектов контроля хранятся в специальном файле - базе эталонов. Файл содержит информацию в формате UTF-8. Местоположение и имя файла указывается при запуске утилиты. Данный файл имеет следующий формат:

[Копировать в буфер обмена](#)

[Нормализованный вид универсального описателя источника информации]  
<объект контроля> = <хеш>

В этом формате [Нормализованный вид универсального описателя источника информации](#) содержит строку, аналогично тому, как она указана в шаблоне источников информации. Однако универсальный описатель источника информации приводится к нормализованному виду. При нормализации выполняются следующие действия:

- Для всех протоколов в конце описания объекта контроля (перед «?») добавляется прямой слеш («/»);
- Для протоколов, описывающих каталоги, в случае отсутствия маски, добавляется маска «\*»;
- Для протоколов, описывающих базы данных, при отсутствии маски добавляется явное указание параметров [config,users](#);
- Для протоколов, описывающих базы данных, маска «\*» заменяется на явное указание [config,users](#);
- Маски лексикографически сортируются;
- Обратные слешы («\») заменяются на прямые («/»);
- Несколько идущих друг за другом слешей (любых) заменяются одним прямым (например, конструкция «//\//» заменяется на «/»).

Строка <объект> = <хеш> - содержит имя объекта контроля и хеш-сумму этого объекта. Таких строк может быть более одной. Если универсальный описатель источника информации содержит указание на конкретный файл, то выражение <объект> отсутствует, и строка сразу начинается с символа «=».

*Пример фрагмента базы эталонов:*

[Копировать в буфер обмена](#)

```
[dbe://C:/1C DB/DB folder/?config]
config = FEC3FC5E46AE98299217D6885B3BE28C4F4D6FB9
[dbe://C:/1C DB/Another DB folder/?config,users]
config = FEC3FC5E46AE98299217D6885B3BE28C4F4D6FB9
users = A244BE3830C2B7075C0BB684896B97A0324984A0
[file://C:/Program Files (x86)/1cv8/8.2.9.356/docs/ru/V8UpdateFrom82Beta.htm]
= 392BC75149AE02565C7E31592EDCD60F00BFA03C
[ndir://C:/Program Files (x86)/1cv8/8.3.5.625/bin/?*]
1cv8.exe = 232F2BCCE6ABB95FD56E3CB3FADE528D851A5D69
1cv8_root.hbk = 025B4C8465D3CD851363B2102478B1CAF2EC444E
1cv8_root.res = 8F48869F82BB222EE25E18502605AC117BB5736A
1cv8_ru.hbk = 81CBB0D5573D8517913053EA88AAADE9785AB443
...
```

#### 4.2.5. Файл отчета

Результат работы утилиты фиксируется в файле отчета. Файл содержит информацию в формате UTF-8. Местоположение и имя файла указывается при запуске утилиты. Однако указанное имя будет преобразовано следующим образом: в имя файла будет добавлено дата и время формирования отчета, например, если указано, что отчета надо поместить в файл [C:\temp\report.rpt](#), то фактическое имя файла будет иметь вид [C:\temp\report-14.02.26-14.07.58.019446.rpt](#).

Сам файл имеет следующий формат:

[Копировать в буфер обмена](#)

```
[.params]
Key=value
[Нормализованный вид универсального описателя источника информации]
<Объект контроля> = <Обнаруженное изменение>
```

Секция [.params](#) присутствует всегда и содержит описание параметров запуска утилиты. В качестве значения [Key](#) могут выступать следующие значения:

- [datetime](#) - дата и время запуска утилиты;
- [workdir](#) - рабочий каталог при запуске утилиты;
- [exepath](#) - путь к исполняемому файлу с утилитой;
- [mode](#) - режим запуска утилиты:
  - [create](#) - режим формирования базы эталонов;
  - [check](#) - режим проверки эталонов.
- [etalon](#) - путь к базе эталонов;
- [report](#) - путь к файлу отчета (как он указан в командной строке запуска утилиты);
- [debug](#) - путь к файлу с отладочной информацией или пустая строка, если не задан вывод отладочной информации;
- [in](#) - путь к файлу с параметрами. Если в командной строке запуска указано более одного параметра [-in](#), то возможно наличие нескольких ключей [in](#) в секции [.params](#);
- [param](#) - все распознанные параметры командной строки запуска утилиты, разделенные запятыми.

Если в каком-либо источнике (из базы эталонов) обнаружены различия между базой эталонов и фактически состоянием файлов на диске, то в файл отчета помещается секция с описанием изменившегося источника и ниже выводятся обнаруженные изменения для объектов контроля. Если объект контроля не изменился - по такому объекту информации не выводится.

Обрабатываются следующие возможные изменения:

- [A](#) - в источнике информации обнаружен новый объект контроля;
- [D](#) - в источнике информации удален объект контроля, который существовал на момент формирования базы эталонов;
- [M](#) - объект контроля модифицирован.

Рассмотрим фрагмент файла отчета:

[Копировать в буфер обмена](#)

```
[dbe://C:/1C DB/DB folder/?config,users]
users = M
[ndir://C:/Program Files (x86)/1cv81/bin/?*]
1CMailV8.dll = D
1CMailV8.dll.32 = A
```

В данном файле отчета зафиксировано, что:

- Для файловой информационной базы, расположенной в каталоге [C:/1C DB/DB folder](#) изменено содержимое таблицы пользователей.

- В каталоге [C:/Program Files \(x86\)/1cv81/bin](#) удален файл [1CmailV8.dll](#) и обнаружен новый файл [1CmailV8.dll.32](#).

## 4.2.6. Использование утилиты

Запуск утилиты осуществляется с помощью командной строки, имеющей следующие параметры:

[Копировать в буфер обмена](#)

```
ci <mode> --in <ШаблонИсточника> --etalon <ФайлБазыЭталонов> --report <ФайлОтчета> --  
debug <ФайлОтладочнаяИнформация> [описание источника]
```

Параметры имеют следующее значение:

*mode*

---

Режим работы утилиты:

- [--version](#), [-v](#) - показать номер версии утилиты контроля целостности.
- [--help](#), [-h](#) - показать справочную информацию по режимам [make](#) и [check](#). В этом случае режим следует указать через пробел после команды получения справочной информации.
- [make](#) - сформировать файл базы эталонов или выполнить обновление данных по каким-либо источникам информации (если база эталонов существует, и в ней присутствуют данные по источнику, который указан при запуске утилиты).
- [check](#) - проверить целостность переданного списка источников по базе эталонов.

Следующие параметры необходимо указывать только при использовании утилиты в режимах [make](#) и [check](#).

*--in, -i*

---

Параметр предназначен для указания пути к шаблону источников информации (см. [здесь](#)). Таких параметров может быть более одного.

Для режимов [make](#) и [check](#) следует указать хотя бы одно описание источника. Это может быть параметр [--in](#) ([-i](#)) или простое указание описания источника (см. ниже параметр [описание источника](#)).

*--etalon, -e*

---

Параметр предназначен для указания полного пути к файлу с базой эталонов (см. [здесь](#)).

Для режимов [make](#) и [check](#) является обязательным параметром.

*--report, -r*

---

Параметр предназначен для указания полного пути к файлу отчетом (см. [здесь](#)).

Для режимов [make](#) и [check](#) является обязательным параметром.

*--debug, -d*

---

Параметр предназначен для указания полного пути к файлу с отладочной информацией. Файл необходим для технического персонала фирмы «1С» в случае расследования некорректного поведения утилиты.

*--method=<значение>, -m=<значение> только для make*

---

Позволяет выбрать алгоритм вычисления хеш-функции для расчета контрольных сумм. У параметра может быть указано одно из следующих значений:

- [sha1](#) - используется алгоритм SHA-1.
- [sha256](#) - используется алгоритм SHA-256. Это значение используется по умолчанию.
- [gost2012](#) - используется алгоритм, описанный в ГОСТ Р 34.11-2012 (512 бит).



Позволяет указать источник информации без указания файла шаблона (в том числе и несколько описаний). Для этого источник следует указывать точно в таком же формате, как и каждая строка списка шаблонов источников информации (см. [здесь](#)). В своей работе утилита использует все описания источников, которые обнаружила при своем запуске (как с помощью параметра `in`, так и с помощью простого указания).

#### 4.2.7. Рекомендации по установке и использованию

Утилита контроля целостности не имеет функций самоконтроля. В связи с этим обеспечение неизменности файлов самой утилиты возложено на системного администратора. В качестве средства обеспечения можно использовать права доступа файловой системы на каталог с утилитой, файлы баз эталонов и файлы отчетов.

В качестве возможного сценария использования можно привести следующий вариант:

1. Утилита устанавливается в каталог, отличающийся от каталога установки системы «1С:Предприятие».
2. Для работы с утилитой создаётся привилегированный пользователь с ограниченными правами, однако, достаточными для работы со всеми объектами контроля в режиме чтения.
3. Права доступа к исполняемым и конфигурационным файлам утилиты ограничиваются созданным пользователем (т. е. любой доступ к файлам любых других пользователей должен быть запрещён).
4. Работать с утилитой следует от имени созданного пользователя.

Вышеперечисленные правила обеспечат высокий уровень защищённости утилиты от несанкционированного доступа и дискредитации.

В этом случае общая схема использования выглядит следующей:

1. Для запуска утилиты необходимо войти в систему от имени привилегированного пользователя. Если утилита запускается с помощью системного планировщика, необходимо обеспечить, чтобы запуск утилиты планировщиком также выполнялся от имени привилегированного пользователя.
2. Выполняется запуск утилиты в одном из необходимых режимов: создания базы эталонов или ее проверки целостности источников.
3. Выполняется анализ отчета о работе утилиты.

Таким образом, применение утилиты (и использование результатов работы утилиты) возможно только для ограниченного круга доверенных лиц, которые знают имя и пароль привилегированного пользователя, созданного для использования утилиты контроля целостности.

### 4.3. Утилита преобразования `cnvdbfl`

Файл базы данных имеет несколько версий внутреннего формата:

1. Версия 8.2.14 - имеет размер внутренней страницы файла базы данных равный 4 096 байт. Размер внутреннего файла не может превышать 4 Гбайта. Более подробно про внутренний файл (см. [здесь](#)).
2. Версия 8.3.8 - размер внутренней страницы файла базы данных может принимать несколько значений: 4 096, 8 192, 16 384, 32 768 и 65 536 байт. Кроме того, формат версии 8.3.8 обеспечивает более оптимальный формат хранения некоторых внутренних данных. Размер внутреннего файла не может превышать 4 Гбайта (при размере страницы в 4 096 байта) и 6 Гбайт (при размере страницы 8 192, 16 384, 32 768 и 65 536 байт). Более подробно про внутренний файл (см. [здесь](#)).

Система «1С:Предприятие» версии 8.3.8 и старше обеспечивает функционирование с файлом `1Cv8.1CD` любого формата без дополнительных действий. Система «1С:Предприятие» версии 8.3.7 и младше обеспечивает функционирование с файлом `1Cv8.1CD` только версии 8.2.14.

Преобразование между двумя форматами возможно либо с помощью операции выгрузки/загрузки данных информационной базы в файл **.dt**, либо с помощью специальной утилиты **cnvdbfl**.

Утилита **cnvdbfl** является утилитой командной строки и позволяет:

1. преобразовывать файлы **1Cv8.1CD** между различными форматами;
2. изменять размер страницы файла для формата 8.3.8.

Для запуска утилиты используется командная строка следующего вида:

---

```
cnvdbfl <ключи> <путь к 1CV8.1CD>
```

[Копировать в буфер обмена](#)

Ключи могут быть следующими:

- **--help** или **-h**

Отображение краткой информации об утилите.

- **--version** или **-v**

Получение версии утилиты.

- **--info** или **-i**

Отображение информации о формате файла **1Cv8.1CD**. Для файлов базы данных, с которыми не работали в системе «1С:Предприятие» версии 8.2.14 и старше, будет выдано диагностическое сообщение о невозможности отобразить информацию о файле.

- **--convert** или **-c**

Выполнить конвертацию файла **1Cv8.1CD**. Если команда используется без указания параметров, то будет выполнена конвертация в формат 8.3.8 с размером страницы равным 8192 байтам.

При исполнении команды следует указать один из следующих параметров:

- **--format=<формат>** или **-f <формат>**

Служит для указания формата, в который следует конвертировать файл **1Cv8.1CD**. **<формат>** может принимать следующие значения:

- **8.2.14** - выполнить конвертацию в формат 8.2.14.
- **8.3.8** - выполнить конвертацию в формат 8.3.8.

- **--page=<размер страницы>** или **-p <размер страницы>**

Служит для указания размера страницы для формата 8.3.8. Значение по умолчанию - 8192. При указании параметра **-f 8.2.14** данный параметр игнорируется. **<размер страницы>** может принимать следующие значения:

- **4096** или **4K**;
- **8192** или **8K**;
- **16384** или **16K**;
- **32768** или **32K**;
- **65536** или **64K**.

Далее представлены примеры использования утилиты.

*Конвертация в формат 8.2.14:*

---

```
cnvdbfl -c -f 8.2.14 c:\temp\1cv8.1cd
```

[Копировать в буфер обмена](#)

*Конвертация в формат 8.3.8 со страницей по умолчанию:*



---

```
cnvdbfl -c -f 8.3.8 c:\temp\1cv8.1cd
```

*Конвертация в формат 8.3.8 со страницей размером 16К:*

[Копировать в буфер обмена](#)

---

```
cnvdbfl -c -f 8.3.8 -p 16k c:\temp\1cv8.1cd
```

*Получить информацию о файле 1Cv8.1CD:*

[Копировать в буфер обмена](#)

---

```
cnvdbfl -i c:\temp\1cv8.1cd
```

## 4.4. Утилита ring

### 4.4.1. Общая информация

Утилита **ring** - это кроссплатформенная консольная (не имеющая графического интерфейса) утилита для управления локальной конфигурацией процессов системы «1С:Предприятия» и для выполнения различных операций, необходимых для поддержки работы системы.

Утилита **ring** имеет модульную архитектуру. Модуль - это набор общей (по смыслу или выполняемым действиям) функциональности, доступ к которой осуществляется с помощью команд. Количество команд в модуле не ограничивается. Команда - это некоторое действие, обладающее некоторым набором параметров (или без таковых). Один вызов утилиты **ring** приводит к выполнению одной команды одного модуля. Одновременно утилита может использовать несколько модулей.

Список установленных и используемых модулей находится в реестре экземпляров модулей **ring-commands.cfg**, формат и расположение которого см. [здесь](#).

Параметры, которые передаются командам, не могут содержать пробелы. Если значение параметра содержит пробел - это значение должно быть заключено в кавычки.

Общая схема вызова команды:

[Копировать в буфер обмена](#)

---

```
ring <module> <command> [--parameter "parameter value"]
```

В приведенной схеме вызова команды:

- **ring** - имя утилиты.
- **<module>** - имя модуля.
- **<command>** - имя используемой команды.
- **--parameter** - параметр и значение параметра для команды. Количество параметров, а также обязательные и необязательные параметры, значения по умолчанию и т. д. определяются для каждой команды индивидуально.

При использовании различных модулей и команд с утилитой **ring** под управлением ОС Windows, необходимо помнить, что при вводе аргументов командной строки вводимые символы не должны требовать более 8 бит. Если для ввода используются языки, в которых существуют символы, не отвечающие этому требованию, рекомендуется подбирать для этих символов соответствующую замену. Например, вместо символа «э» можно попробовать использовать символ «а» или «е».

### 4.4.2. Системные требования

Работа утилиты **ring** поддерживается на тех же операционных системах, что и собственно «1С:Предприятие» (см. [здесь](#)). Утилита **ring** доступна в 32- и 64-разрядных версиях.

Дополнительно для работы утилиты **ring** требуются:

- Java 8 и выше (<http://www.java.com/ru/download/>).

- При работе под управлением ОС Linux необходимо, чтобы GNU Awk был установлен и использовался по умолчанию.

### 4.4.3. Установка утилиты

Установка утилиты выполняется при установке других продуктов, которые требуют данную утилиту. Другие продукты могут являться модулями утилиты `ring` или включать в себя модули для этой утилиты.

Утилита `ring` без установленных модулей не имеет какой-либо практической ценности, поэтому установка отдельно утилиты `ring` не поддерживается.

### 4.4.4. Рекомендации по использованию

#### 4.4.4.1. Установка языка вывода утилиты

Утилита `ring` выводит информацию на языке, определенном операционной системой. Информацию о способе смены языка интерфейса можно получить в документации на используемую операционную систему.

Если необходимо, чтобы утилита `ring` выводила информацию на языке, отличном от языка в настройках операционной системы, то сделать это можно путем указания в переменной окружения `RING_OPTS` параметра `-Duser.country=ru -Duser.language=RU`. Если переменная `RING_OPTS` уже имеет значение, то к значению следует добавить пробел и после него указанную строку.

Пример установки переменной окружения с помощью командной строки:

- Для ОС Windows:

[Копировать в буфер обмена](#)

```
set RING_OPTS=-Duser.country=ru -Duser.language=RU
```

- Для ОС Linux (командный процессор `bash`):

[Копировать в буфер обмена](#)

```
export RING_OPTS=-Duser.country=ru -Duser.language=RU
```

Утилита поддерживает два языка вывода: русский и английский.

#### 4.4.4.2. Некорректный вывод утилиты

Обязательно установите для терминала шрифт Consolas или другой TrueType-шрифт с поддержкой символов Unicode. Проверьте, что нужный язык выбран в операционной системе в качестве языка для программ, не поддерживающих Unicode. Изменение данной настройки вступает в силу после перезагрузки системы.

Если изменить настройки операционной системы не представляется возможным, то при использовании шрифта Consolas попробуйте до запуска `ring` выполнить команду по смене кодовой страницы, указав кодовую страницу, подходящую для нужного языка, например:

[Копировать в буфер обмена](#)

```
chcp 1251
```

#### 4.4.4.3. Задание кодировки при переадресации вывода утилиты в файл

При работе под управлением ОС Windows возможна ситуация, когда переадресация вывода утилиты `ring` в файл приводит к тому, что содержимое файла содержит нечитаемые символы. Такая ситуация возникает при несоответствии настройки языка для программ, не поддерживающих Unicode, и языка в интерфейсе системы. Существует несколько способов получить данные в файле в читаемом виде:

- Установить язык для программ, не поддерживающих Unicode в соответствии с языком интерфейса, подробнее см. [здесь](#).
- Указать кодировку UTF-8 используя переменную окружения `RING_OPTS`, подробнее см. [здесь](#).

## 4.5. Утилита лицензирования

### 4.5.1. Общая информация

Утилита лицензирования предназначена для выполнения следующих задач:

- первичное получение лицензии;
- повторное получение и обновление лицензии;
- проверка соответствия файла лицензии текущему компьютеру;
- вывод списка лицензий;
- получение информации о лицензии;
- удаление лицензии;
- обновление лицензии.

При описании утилиты лицензирования используется понятие «хранилище лицензий». Под этим термином следует понимать каталог, в котором располагаются файлы с активированными программными лицензиями. В зависимости от используемой ОС, утилита лицензирования по умолчанию использует следующие каталоги в качестве хранилища лицензий:

- ОС Windows: `%ALLUSERSPROFILE%\1C\licenses` (`%ProgramData%\1C\licenses` для ОС Windows Vista и старше).
- ОС Linux: `/var/1C/licenses`.

Полный список каталогов, где могут располагаться программные лицензии см. [здесь](#).

При отображении лицензий используется унифицированное представление вида: пинкод-номер. Так, если имеется программная лицензия, для которой возможно использование пинкода 123-456-789-012 и эта лицензия используется для продукта с регистрационным номером 8000314159, то данная лицензия будет отображаться следующим образом: 123456789012-8000314159. Это же имя следует указывать в том случае, если какая-либо команда утилиты лицензирования требует в качестве параметра имя лицензии.

Для активации программной лицензии можно использовать два варианта:

1. Использование команды `activate`. В данном случае активация происходит сразу, с использованием веб-сервиса центра лицензирования, без формирования промежуточных файлов.
2. Использование последовательности команд `prepare-request`, `acquire` и `generate`. В этом случае вначале подготавливается файл запроса к центру лицензирования (команда `prepare-request`), затем выполняется запрос к центру лицензирования (с помощью электронной почты) и получается ответ центра лицензирования (в виде файла). Также имеется возможность выполнить формирование файла ответа от центра лицензирования через веб-сервис, без использования электронной почты (команда `acquire`). Затем на основании запроса и ответа получается файл лицензии (команда `generate`).

### 4.5.2. Системные требования

Утилита лицензирования, фактически, является отдельным модулем, требующим для своего исполнения утилиту `ring`. Модуль называется `license`. С точки зрения выполняемых действий, «модуль `license`» и «утилита лицензирования», являются взаимозаменяемыми понятиями. С технической точки зрения «утилита лицензирования» - это утилита `ring` и модуль `license`, а «модуль `license`» - это отдельный модуль, требующий для своей работы утилиту `ring`.

Для работы модуля `license` требуется установленная утилита `ring`. Модуль `license` доступен в 32- и 64-разрядных версиях.

Для работы модуля `license` требуются:

- Утилита **ring** версии 0.8.2 и выше.
- JCE для Java 8 (<http://www.oracle.com/technetwork/java/javase/downloads/ice8-download-2133166.html>) или корректная настройка конфигурационного файла Java (в зависимости от версии Java).
- Для ОС Windows: система Windows Script Host, работающая на компьютере, на котором запускается модуль **license**.

### 4.5.3. Установка модуля

#### 4.5.3.1. Установка модуля

Дистрибутив утилиты лицензирования поставляется совместно с дистрибутивом системы «1С:Предприятие» в виде набора файлов. Эти файлы располагаются в каталоге **license-tools** дистрибутива платформы. В состав каталога входят файлы с нужными пакетами и утилиты **1ce-installer** (программа установки с графическим интерфейсом) и **1ce-installer-cli** (программа установки с интерфейсом командной строки). Разрядность и поддерживаемая операционная система для утилиты лицензирования совпадает с разрядностью и поддерживаемой операционной системой дистрибутива платформы «1С:Предприятие».

Если дистрибутив платформы скачивается с сервиса «1С:Обновление программ» (<https://releases.1c.ru/>), то для получения доступа к установочным файлам утилиты лицензирования, скачанный архив необходимо распаковать на локальном диске (вместе со всеми вложенными подкаталогами).

Версию утилиты лицензирования, утилиты **ring** и программы установки, можно определить в именах файлов. Например, для файлов с такими именами:

[Копировать в буфер обмена](#)

```
1c-enterprise-installer-1.6.0+4-Windows-anyarch.elc.car  
1c-enterprise-license-tools-0.12.0+3-Windows-x86_64.elc.car  
1c-enterprise-license-tools-0.12.0+3-Windows-x86_64.elc.dar  
1c-enterprise-license-tools-0.12.0+3-Windows-x86_64.elc.par  
1c-enterprise-ring-0.17.0+1-Windows-x86_64.elc.car
```

Можно определить следующие версии (в списке выделены полужирным шрифтом):

- Версия программы установки: **1.6.0.4**.
- Версия утилиты **ring**: **0.17.0.1**.
- Версия утилиты лицензирования: **0.12.0.3**.

Для установки утилиты необходимо перейти в каталог **license-tools** распакованного дистрибутива платформы и запустить одну из программ установки. Описание работы с программой установки (как в графической, так и в консольной версии) можно получить на ИТС (<https://its.1c.ru/db/inst10doc>).

После установки модуля можно проверить, что он зарегистрировался в реестре экземпляров модулей (см. [здесь](#)).

#### 4.5.3.2. Настройка Java

Для корректной работы утилиты лицензирования, в зависимости от версии используемой Java, должны быть выполнены некоторые дополнительные настройки:

- Версия Java младше 1.8 update 151 - необходимо установить JCE.
- Версия Java равна или старше 1.8 update 151 - возможно два варианта:
  - Установить в файле **java.security** параметр **crypto.policy** в значение **unlimited** (внести в этот конфигурационный файл строку **crypto.policy=unlimited** или убрать комментарий у этой строки).
  - Установить JCE.

- Версия Java 9 и выше - установить в файле [java.security](#) параметр `crypto.policy` в значение `unlimited` (внести в этот конфигурационный файл строку `crypto.policy=unlimited` или убрать комментарий у этой строки).

В зависимости от версии Java и используемого варианта Java, конфигурационный файл [java.security](#) располагается в различных местах:

- Java 1.8:
  - JDK - `%JAVA_HOME%\jre\lib\security`.
  - JRE - `%JAVA_HOME%\lib\security`.
- Java 9 и выше: `%JAVA_HOME%\conf\security`.

## 4.5.4. Команды модуля

### 4.5.4.1. Команда activate

Активация лицензии с использованием веб-сервиса центра лицензирования.

Командная строка:

[Копировать в буфер обмена](#)

```
ring license activate --first-name <имя> --middle-name <отчество> --last-name <фамилия>
[--email <email>] --company <компания> --country <страна> --zip-code <индекс> [--region
<область> --district <район>] --town <город> --street <улица> --house <дом> --building
<строение> --apartment <квартира> --serial <серийныйНомер> --pin <пинкод> [--previous-
pin <предыдущийПинкод>] [--path <хранилище>] [--validate] [--conf-location <путь>] [--
send-statistics <режим>]
```

Описание параметров:

- `first-name` - имя владельца лицензии. При указании параметра `company`, данный параметр является необязательным.
- `middle-name` - отчество владельца лицензии. При указании параметра `company`, данный параметр является необязательным.
- `last-name` - фамилия владельца лицензии. При указании параметра `company`, данный параметр является необязательным.
- `email` - электронная почта владельца лицензии.
- `company` - организация владельца лицензии. При указании параметров `first-name`, `middle-name`, `last-name`, данный параметр является необязательным. Требуется не менее 5 символов, при этом не должно быть более 3 одинаковых символов подряд.
- `country` - страна регистрации. Не может быть пустым.
- `zip-code` - индекс. Не может быть пустым.
- `region` - область/республика/край.
- `district` - район.
- `town` - город. Не может быть пустым.
- `street` - улица. Не может быть пустым.
- `house` - номер дома. При указании параметров `building` или `apartment`, данный параметр является необязательным. Не может быть пустым.
- `building` - строение. При указании параметров `house` или `apartment`, данный параметр является необязательным. Не может быть пустым.
- `apartment` - квартира. При указании параметров `house` или `building`, данный параметр является необязательным. Не может быть пустым.

- `serial` - серийный номер программного продукта.
- `pin` - пинкод, используемый при активации лицензии.
- `previous-pin` - при повторной активации лицензии в данном параметре указывается пинкод, который использовался при первичной активации лицензии. Не должен совпадать со значением параметра `pin`.
- `path` - указывает путь к хранилищу лицензий, если оно отличается от пути по умолчанию.
- `validate` - если указан, то выполнение команды будет завершено с ошибкой, если при попытке получения какого-либо из ключевых параметров возникла ошибка времени исполнения. В случае если параметр не указан, возникновение ошибки при получении какого-либо ключевого параметра не будет препятствовать успешной активации лицензии. Однако, поля лицензии, соответствующие неполученным параметрам, будут заполнены пустыми значениями, что приведет к невозможности дальнейшего использования активированной лицензии.
- `conf-location` - позволяет указать каталог, в котором расположен конфигурационный файл `conf.cfg`. В данном файле анализируется параметр `ExternalResourcesMode`, в соответствии с которым выбирается адрес используемого центра лицензирования.

Если параметр не задан или в указанном каталоге не обнаружен файл `conf.cfg`, то выполняется поиск этого файла в следующих каталогах:

- Для ОС Windows:
  - 32-разрядная система в 64-разрядной ОС: `%PROGRAMFILES(x86)%\1cv8\conf`.
  - В остальных случаях: `%PROGRAMFILES%\1cv8\conf`.
- Для ОС Linux:
  - Каталог `~/1cv8/1C/1cv8/conf` (~ - домашний каталог пользователя, от имени которого работает сервер «1С:Предприятия»).
- `send-statistics` - управляет отправкой статистической информации в сервис 1С:Пульт (адрес `pult.1c.com`, порт `443`). Параметр `<режим>` может принимать следующие значения:
  - `true` - информация отправляется. Значение по умолчанию.
  - `false` - информация не отправляется.

Перечень отправляемой информации см. [здесь](#).

#### Результат работы:

---

При успешной регистрации, сервер возвращает файл лицензии, который помещается в указанное хранилище. При повторном вводе идентичных данных (включая данные комплекта) на той же машине, генерируется подобный файл лицензии.

В случае успешного завершения код возврата утилиты равен 0. В противном случае код возврата отличается от 0 и в стандартный поток вывода формируется сообщение об ошибке.

Смотри также:

- Файл `conf.cfg` (см. [здесь](#)).

#### 4.5.4.2. Команда `prepare-request`

В том случае, если активировать программную лицензию с использованием веб-сервиса центра лицензирования не представляется возможным, имеется возможность активировать лицензию с помощью электронного письма в центр лицензирования. Для этого используются команды `prepare-request`, `acquire` и `generate` в указанном порядке.

Команда `prepare-request` выполняет подготовку файла для передачи в центр лицензирования.



---

*Командная строка:*

---

[Копировать в буфер обмена](#)

```
ring license prepare-request --first-name <имя> --middle-name <отчество> --last-name <фамилия> [--email <email>] --company <компания> --country <страна> --zip-code <индекс> [--region <область> --district <район>] --town <город> --street <улица> --house <дом> --building <строение> --apartment <квартира> --serial <серийныйНомер> --pin <пинкод> [--previous-pin <предыдущийПинкод> --request <файл>] --validate [--send-statistics <режим>]
```

---

*Описание параметров:*

---

- **first-name** - имя владельца лицензии. При указании параметра **company**, данный параметр является необязательным.
- **middle-name** - отчество владельца лицензии. При указании параметра **company**, данный параметр является необязательным.
- **last-name** - фамилия владельца лицензии. При указании параметра **company**, данный параметр является необязательным.
- **email** - электронная почта владельца лицензии.
- **company** - организация владельца лицензии. При указании параметров **first-name**, **middle-name**, **last-name**, данный параметр является необязательным. Требуется не менее 5 символов, при этом не должно быть более 3 одинаковых символов подряд.
- **country** - страна регистрации. Не может быть пустым.
- **zip-code** - индекс. Не может быть пустым.
- **region** - область/республика/край.
- **district** - район.
- **town** - город. Не может быть пустым.
- **street** - улица. Не может быть пустым.
- **house** - номер дома. При указании параметров **building** или **apartment**, данный параметр является необязательным. Не может быть пустым.
- **building** - строение. При указании параметров **house** или **apartment**, данный параметр является необязательным. Не может быть пустым.
- **apartment** - квартира. При указании параметров **house** или **building**, данный параметр является необязательным. Не может быть пустым.
- **serial** - серийный номер программного продукта.
- **pin** - пинкод, используемый при активации лицензии.
- **previous-pin** - при повторной активации лицензии в данном параметре указывается пинкод, который использовался при первичной активации лицензии. Не должен совпадать со значением параметра **pin**.
- **request** - указывает полный путь к файлу, в который будет помещена информация для передачи в центр лицензирования. Если не указан, то текст запроса в центр лицензирования будет выведен в стандартный поток вывода.
- **validate** - если указан, то выполнение команды будет завершено с ошибкой, если при попытке получения какого-либо из ключевых параметров возникла ошибка времени исполнения. В случае если параметр не указан, возникновение ошибки при получении какого-либо ключевого параметра не будет препятствовать успешной активации лицензии. Однако, поля лицензии, соответствующие неполученным параметрам, будут заполнены пустыми значениями, что приведет к невозможности дальнейшего использования активированной лицензии.

- `send-statistics` - управляет отправкой статистической информации в сервис 1С:Пульт (адрес [pult.1c.com](http://pult.1c.com), порт 443). Параметр `<режим>` может принимать следующие значения:

- `true` - информация отправляется. Значение по умолчанию.
- `false` - информация не отправляется.

Перечень отправляемой информации см. [здесь](#).

---

#### Результат работы:

Формируется содержимое файла запроса в центр лицензирования (в файле или в стандартном потоке вывода).

В случае успешного завершения код возврата утилиты равен 0. В противном случае код возврата отличается от 0 и в стандартный поток вывода формируется сообщение об ошибке.

#### 4.5.4.3. Команда `acquire`

Команда предназначена для формирования файла ответа по ранее сформированному файлу запроса, с использованием веб-сервиса центра лицензирования.

---

#### Командная строка:

[Копировать в буфер обмена](#)

```
ring license acquire [--request <ФайлЗапроса>] [--response <ФайлОтвета>] [--conf-  
location] [--send-statistics <режим>]
```

---

#### Описание параметров:

- `request` - полное имя к файлу с запросом к центру лицензирования. Если параметр не указан, то содержимое файла запроса ожидается со стандартного потока ввода.
- `response` - полное имя файла, в которое будет помещен ответ центра лицензирования. Если параметр не указан, то содержимое файла ответа будет выведено в стандартный поток вывода.
- `conf-location` - позволяет указать каталог, в котором расположен конфигурационный файл `conf.cfg`. В данном файле анализируется параметр `ExternalResourcesMode`, в соответствии с которым выбирается адрес используемого центра лицензирования.

Если параметр не задан или в указанном каталоге не обнаружен файл `conf.cfg`, то выполняется поиск этого файла в следующих каталогах:

- Для ОС Windows:
  - 32-разрядная система в 64-разрядной ОС: `%PROGRAMFILES(x86)%\1cv8\conf`.
  - В остальных случаях: `%PROGRAMFILES%\1cv8\conf`.
- Для ОС Linux:
  - Каталог `~/.1cv8/1C/1cv8/conf` (~ - домашний каталог пользователя, от имени которого работает сервер «1С:Предприятия»).
- `send-statistics` - управляет отправкой статистической информации в сервис 1С:Пульт (адрес [pult.1c.com](http://pult.1c.com), порт 443). Параметр `<режим>` может принимать следующие значения:
  - `true` - информация отправляется. Значение по умолчанию.
  - `false` - информация не отправляется.

Перечень отправляемой информации см. [здесь](#).

---

#### Результат работы:

Ответ центра лицензирования.

В случае успешного завершения код возврата утилиты равен 0. В противном случае код возврата отличается от 0 и в стандартный поток вывода формируется сообщение об ошибке.

#### 4.5.4.4. Команда generate

Команда предназначена для формирования файла лицензии по данным запроса к центру лицензирования и его (центра) ответа.

Командная строка:

[Копировать в буфер обмена](#)

```
ring license generate --license <файлЛицензии> --request <ФайлЗапроса> --response  
<ФайлОтвета> [--send-statistics <режим>]
```

Описание параметров:

- **license** - полное имя к файлу с получившейся лицензией. Если параметр не указан, то содержимое файла активированной лицензии выводится в стандартный поток вывода.
- **request** - полное имя к файлу с запросом к центру лицензирования.
- **response** - полное имя файла, в которое будет помещен ответ центра лицензирования.
- **send-statistics** - управляет отправкой статистической информации в сервис 1С:Пульт (адрес [pult.1c.com](http://pult.1c.com), порт 443). Параметр **<режим>** может принимать следующие значения:
  - **true** - информация отправляется. Значение по умолчанию.
  - **false** - информация не отправляется.

Перечень отправляемой информации см. [здесь](#).

Результат работы:

Файл с активированной лицензией.

В случае успешного завершения код возврата утилиты равен 0. В противном случае код возврата отличается от 0 и в стандартный поток вывода формируется сообщение об ошибке.

#### 4.5.4.5. Команда list

Команда предназначена для отображения списка лицензий в хранилище лицензий.

Командная строка:

[Копировать в буфер обмена](#)

```
ring license list [--path <хранилище>] [--send-statistics <режим>]
```

Описание параметров:

- **path** - указывает путь к хранилищу лицензий, если оно отличается от пути по умолчанию.
- **send-statistics** - управляет отправкой статистической информации в сервис 1С:Пульт (адрес [pult.1c.com](http://pult.1c.com), порт 443). Параметр **<режим>** может принимать следующие значения:
  - **true** - информация отправляется. Значение по умолчанию.
  - **false** - информация не отправляется.

Перечень отправляемой информации см. [здесь](#).

Результат работы:

Список имен обнаруженных лицензий.

#### 4.5.4.6. Команда info

Выводит информацию об указанной лицензии.

---

**Командная строка:**

---

[Копировать в буфер обмена](#)

```
ring license info [--name <имя>] [--path <хранилище>] [--send-statistics <режим>]
```

---

**Описание параметров:**

---

- **name** - имя лицензии, для которой ожидается получение информации. Если данный параметр отсутствует, то система ожидает со стандартного потока ввода содержимое (не имя!) файла с активированной лицензией.
- **path** - указывает путь к хранилищу лицензий, если оно отличается от пути по умолчанию.
- **send-statistics** - управляет отправкой статистической информации в сервис 1С:Пульт (адрес [pult.1c.com](http://pult.1c.com), порт [443](#)). Параметр **<режим>** может принимать следующие значения:
  - **true** - информация отправляется. Значение по умолчанию.
  - **false** - информация не отправляется.

Перечень отправляемой информации см. [здесь](#).

---

**Результат работы:**

---

Информация о лицензии в следующем формате:

[Копировать в буфер обмена](#)

Информация о пользователе:

Имя :  
Отчество :  
Фамилия :  
Страна :  
Индекс :  
Город :  
Улица :  
Квартира/офис :

Информация о продукте:

Дата комплектации :  
Описание :  
Регистрационный номер :  
Код продукта :

При выводе не выполняется соответствие текущего оборудования компьютера и оборудования, использованного при активизации лицензии.

#### 4.5.4.7. Команда validate

Проверяет соответствие оборудования на компьютере оборудованию, которое было зафиксировано в момент активации лицензии.

---

**Командная строка:**

---

[Копировать в буфер обмена](#)

```
ring license validate [--name <имя>] [--path <хранилище>] [--send-statistics <режим>]
```

---

**Описание параметров:**

---

- **name** - имя лицензии, для которой ожидается получение информации. Если данный параметр отсутствует, то система ожидает со стандартного потока ввода содержимое (не имя!) файла с активированной лицензией.
- **path** - указывает путь к хранилищу лицензий, если оно отличается от пути по умолчанию.
- **send-statistics** - управляет отправкой статистической информации в сервис 1С:Пульт (адрес [pult.1c.com](http://pult.1c.com), порт [443](#)). Параметр **<режим>** может принимать следующие значения:
  - **true** - информация отправляется. Значение по умолчанию.
  - **false** - информация не отправляется.

Перечень отправляемой информации см. [здесь](#).

---

#### Результат работы:

---

Либо информация о том, что оборудование компьютера соответствует оборудованию, зафиксированному в момент активации лицензии, либо отображает перечень различий (если перечень оборудования различается).

В случае успешного завершения код возврата утилиты равен 0. В противном случае код возврата отличается от 0 и в стандартный поток вывода формируется сообщение об ошибке. Отдельные ситуации выделяются специальными кодами ошибок:

- 1 - изменены параметры компьютера;
- 2 - ошибка проверки цифровой подписи лицензии.

#### 4.5.4.8. Команда put

Помещает выбранный файл с активированной лицензией в указанное хранилище лицензий.

---

#### Командная строка:

---

[Копировать в буфер обмена](#)

```
ring license put --license <полноеИмя> [--path <хранилище>] [--send-statistics <режим>]
```

---

#### Описание параметров:

---

- **license** - полный путь к файлу активированной лицензии, который будет помещен в хранилище лицензий.
- **path** - указывает путь к хранилищу лицензий, если оно отличается от пути по умолчанию.
- **send-statistics** - управляет отправкой статистической информации в сервис 1С:Пульт (адрес [pult.1c.com](https://pult.1c.com), порт 443). Параметр **<режим>** может принимать следующие значения:
  - **true** - информация отправляется. Значение по умолчанию.
  - **false** - информация не отправляется.

Перечень отправляемой информации см. [здесь](#).

---

#### Результат работы:

---

Помещает указанный файл с активированной лицензией в указанное (явно или неявно) хранилище лицензий.

В случае успешного завершения код возврата утилиты равен 0. В противном случае код возврата отличается от 0.

#### 4.5.4.9. Команда get

Получает файл с указанной активированной лицензией в указанном хранилище лицензий и записывает в файл.

---

#### Командная строка:

---

[Копировать в буфер обмена](#)

```
ring license get -name <имя> [--license <полноеИмя>] [--path <хранилище>] [--send-statistics <режим>]
```

---

#### Описание параметров:

---

- **name** - имя лицензии, для которой ожидается получение информации.
- **license** - полный путь к файлу, в который будет записана получаемая лицензия. Если не указано - содержимое файла с лицензией выводится в стандартный поток вывода.
- **path** - указывает путь к хранилищу лицензий, если оно отличается от пути по умолчанию.

• `send-statistics` - управляет отправкой статистической информации в сервис 1С:Пульт (адрес [pult.1c.com](https://pult.1c.com), порт 443). Параметр `<режим>` может принимать следующие значения:

- `true` - информация отправляется. Значение по умолчанию.
- `false` - информация не отправляется.

Перечень отправляемой информации см. [здесь](#).

---

#### Результат работы:

Получает из хранилища файл с активированной лицензией и помещает его в файл с указанным именем.

В случае успешного завершения код возврата утилиты равен 0. В противном случае код возврата отличается от 0.

#### 4.5.4.10. Команда `remove`

Удаляет указанную лицензию из указанного хранилища лицензий.

---

#### Командная строка:

[Копировать в буфер обмена](#)

```
ring license remove -name <имя> [--path <хранилище>] [--all] [--send-statistics <режим>]
```

---

#### Описание параметров:

- `name` - имя лицензии, которую требуется удалить из хранилища.
- `path` - указывает путь к хранилищу лицензий, если оно отличается от пути по умолчанию.
- `all` - удалить все лицензии с заданным именем в хранилище.
- `send-statistics` - управляет отправкой статистической информации в сервис 1С:Пульт (адрес [pult.1c.com](https://pult.1c.com), порт 443). Параметр `<режим>` может принимать следующие значения:
  - `true` - информация отправляется. Значение по умолчанию.
  - `false` - информация не отправляется.

Перечень отправляемой информации см. [здесь](#).

---

#### Результат работы:

Удаленная лицензия в хранилище лицензий.

В случае успешного завершения код возврата утилиты равен 0. В противном случае код возврата отличается от 0.

#### 4.5.4.11. Команда `update`

Выполняет обновление (переполучение) всех лицензий из хранилища лицензий. Под обновлением лицензии понимается повторное получение лицензии в центре лицензирования с теми же параметрами: регистрационный номер, пинкод, ключевые параметры. Существующие до обновления файлы с активированными программными лицензиями сохраняются с расширением `.oldlic` с целью резервного копирования.

---

#### Командная строка:

[Копировать в буфер обмена](#)

```
ring license update --conf-location <КонфигурационныйФайл> [--force <значение>] [--path <хранилище>] [--validate] [--send-statistics <режим>]
```

---

#### Описание параметров:



- `conf-location` - каталог расположения конфигурационных файлов платформы. Используется для поиска конфигурационного файла `conf.cfg`, в котором анализируется параметр `ExternalResourcesMode`. Если параметр не указан или в указанном каталоге отсутствует файл `conf.cfg`, то будет выполнена попытка найти файл `conf.cfg` в следующих каталогах:

- Для ОС Windows:

- 32-разрядная система в 64-разрядной ОС: `%PROGRAMFILES(x86)%\1cv8\conf`.
- В остальных случаях: `%PROGRAMFILES%\1cv8\conf`.

- Для ОС Linux:

- Каталог `~/.1cv8/1C/1cv8/conf` (~ - домашний каталог пользователя, от имени которого запускается утилита лицензирования).

- `force` - определяет поведение утилиты лицензирования, если при обновлении лицензии произошли ошибки. Параметр может принимать следующие значение:

- `true` - ошибка игнорируется и файл, при обработке которого произошла ошибка, пропускается. Обработка продолжается.
- `false` - обработка файлов прерывается, все лицензии восстанавливаются из резервных копий (значение по умолчанию).

- `path` - указывает путь к хранилищу лицензий, если оно отличается от пути по умолчанию.

- `validate` - указывает, следует ли проверять аппаратные данные, полученные от системы. Параметр может принимать следующие значения:

- `true` - проверять.
- `false` - не проверять (значение по умолчанию).

- `send-statistics` - управляет отправкой статистической информации в сервис 1С:Пульт (адрес [pult.1c.com](https://pult.1c.com), порт 443). Параметр `<режим>` может принимать следующие значения:

- `true` - информация отправляется. Значение по умолчанию.
- `false` - информация не отправляется.

Перечень отправляемой информации см. [здесь](#).

---

#### Результат работы:

Обновлены все лицензии в хранилище лицензий.

В случае успешного завершения код возврата утилиты равен 0. В противном случае код возврата отличается от 0.

#### 4.5.4.12. Информация для сервиса 1С:Пульт

В сервис 1С:Пульт отправляются следующие данные:

- Тип операционной системы: Windows или Linux.
- Полное имя операционной системы.
- Архитектура компьютера, на котором запустили утилиту.
- Версия запущенной утилиты лицензирования.
- Архитектура запущенной утилиты лицензирования.
- Версия Java, под управлением которой работает утилита.
- Название исполняемой команды утилиты лицензирования.

- Уникальный идентификатор установки, сгенерированный на основе следующей информации (хеш в формате UUID):
  - MAC-адрес компьютера;
  - версия утилиты;
  - архитектура утилиты.
- Номер пакета данных.
- Уникальный идентификатор компьютера, сгенерированный на основе MAC-адреса сервера (хеш в формате UUID).
- Дата и время выполнения команды.

## 4.6. Утилита административной консоли 1cv8a

Утилита административной консоли ([1cv8a](#)) предназначена для ускорения проверки и исправления определенных проблем:

- Тестирование и исправление таблиц узлов планов обмена.
- Тестирование и исправление хеш-полей таблиц информационной базы.

Утилита административной консоли позволяет исправлять некоторые проблемы, возникающие с информационной базой, без запуска конфигуратора и за более короткое время. Это обусловлено тем, что утилита административной консоли занимается обработкой только проблемных объектов и при этом выполняется ограниченный набор исправлений.

Для запуска утилиты используется командная строка следующего вида:

---

```
1cv8a <режимы> <ключи>
```

[Копировать в буфер обмена](#)

Режимы могут быть следующими:

- [help \[режим\]](#) или [h \[режим\]](#)

Отображение краткой информации об утилите или какому-либо режиму работы.

- [ib-check-and-repair](#) или [ibcr](#)

Выполнение операции тестирования и исправления данных.

При исполнении команды можно указывать следующие параметры:

- [--repair](#) или [-r](#)

Указывает на необходимость выполнить исправление найденных ошибок.

- [--exchange-plan-integrity](#) или [-epi](#)

Выполнять тестирование и исправление таблиц узлов планов обмена.

- [--exchange-plan-integrity-outFile=<имя XML-файла>](#) или [-epiof <имя XML-файла>](#)

Указывает имя XML-файла для сохранения состояния записей [ЭтотУзел](#) планов обмена. Параметр является обязательным при выполнении команды с параметром [--exchange-plan-integrity](#).

- [--dimhash-integrity](#) или [-dhi](#)

Выполняет пересчет итогов для устранения некорректного расчета итогов для регистров накопления и бухгалтерии, если среди измерений есть хотя-бы одно измерение с типом [Строка](#) и в состав индекса по измерениям входят более 16 полей базы данных. Пересчет итога выполняется только в том случае, если одновременно с параметром [--dimhash-integrity](#) используется параметр [--repair](#). Если параметр [--repair](#) не указан - выполняется

только проверка того, что в информационной базе есть некорректные итоги регистров накопления и бухгалтерии.

- `--connection-string=<string>`

Указывает строку соединения с информационной базой.

- `--file=<path>` или `-f <path>`

Указывает путь к каталогу информационной базы в файловом варианте.

- `--server=<server:port/infobase>` или `-s <server:port/infobase>`

Указывает параметры подключения к информационной базе в клиент-серверном варианте.

- `--user=<name>` или `-u <name>` или `-n <name>`

Указывает имя пользователя, от имени которого будет выполняться подключение к информационной базе.

- `--password=<password>` или `-p <password>`

Указывает пароль пользователя информационной базы, от имени которого будет выполняться подключение к информационной базе.

- `--unlock-code=<code>`

Указывает код разрешения соединения с информационной базой (если задан).

- `--security-level=<level>`

Указывает уровень защищенности соединения с сервером.

- `--verbose`

Указывает на необходимость подробного вывода сообщений о ходе выполнения.

## 4.7. Работа конфигуратора в режиме агента

### 4.7.1. Общая информация

**Режим агента** - специальный вариант пакетного режима запуска конфигуратора, в котором он (конфигуратор) выполняет функции SSH- и SFTP-сервера и принимает команды по этим протоколам.

При работе в режиме агента, конфигуратор в один момент времени может работать только с одной информационной базой. Если нужно одновременно работать с несколькими информационными базами - необходимо запускать несколько конфигураторов в режиме агента.

Общая схема работы в режиме агента выглядит следующим образом:

1. Выполняется запуск конфигуратора в режиме агента.
2. С удаленного компьютера выполняется подключение к агенту.
3. С помощью SFTP-клиента выполняется обмен файлами с агентом.
4. С помощью SSH-клиента выполняется передача команд агенту и получение результата работы.

В режиме агента конфигуратор предоставляет определенный набор команд, который будет рассмотрен в данном разделе. Все команды исполняются интерактивно и синхронно. При выполнении команды не предоставляется возможность узнать состояние выполнения. Все сообщения, которые формируются после выполнения команд, полностью аналогичны таковым при выполнении аналогичных команд пакетного режима запуска конфигуратора.

```
login as: adminadmin
admin@localhost's password:
1C:Enterprise 8.3 1C Designer Shell © 1C LLC 1996-2018
designer>
```

Результат работы команд в режиме агента могут быть представлены в двух форматах: в текстовом формате (по умолчанию) и в формате JSON-сообщений. При работе в формате JSON-сообщений, сервер всегда возвращает массив с информацией о результате работы. Для установки формата результата работы предназначена специальная команда. В конце обработки всех команд агент конфигулятора отправляет сообщение с типом `success`, `error`, `cancel` или `question`. Команда `help` отправляет в конце сообщение с типом `success`, а при вводе неизвестной команды - сообщение с типом `error`.

Для того чтобы запустить режим агента, следует выполнить команду следующего вида:

[Копировать в буфер обмена](#)

```
lcv8 DESIGNER <информационная база> /AgentMode [параметры режима агента] [/Visible].
```

При указании командной строки запуска следует учитывать следующие особенности:

- Для указания информационной базы может использоваться любой из доступных способов: параметры `/IBName`, `/F` или `/S`. При этом параметры, определяющие имя пользователя и пароль доступа к информационной базе, указывать не имеет смысла, эти параметры будут проигнорированы. Логин и пароль будет необходимо ввести в ответ на соответствующие запросы в консоли удаленного доступа. Также следует помнить, что для подключения к информационной базе, которая будет использоваться в режиме агента, не рекомендуется использовать пользователя, имя которого задано с использованием национальных символов.
- В том случае, если в командной строке будет обнаружен параметр `/AgentMode`, конфигуратор будет игнорировать все параметры, кроме: параметров режима агента; параметров, указывающих информационную базу; параметра `/Visible`.
- Указание параметра `/Visible` приводит к появлению на экране специального окна, с помощью которого можно завершить работу режима агента. Если запуск выполняется без этого параметра, то завершить работу в режиме агента можно только с помощью командной строки удаленного доступа.

К конфигуратору, работающему в режиме агента, может быть подключен один клиент по протоколу SSH и несколько клиентов по протоколу SFTP.

Смотри также:

- Описание работы с JSON (см. стр. см. [здесь](#)).
- Команды работы в режиме агента (см. [здесь](#)).

## 4.7.2. Список команд

В режиме агента предоставляются следующие команды:

- Команда `help` (подробнее см. [здесь](#)).
- Группа команд `common` - общие команды (подробнее см. [здесь](#)).
- Группа команд `options` - команды работы с настройками (подробнее см. [здесь](#)).
- Группа команд `config` - команды редактирования конфигурации (подробнее см. [здесь](#)).
- Группа команд `infobase-tools` - сервисные команды работы с информационной базой (подробнее см. [здесь](#)).

Более подробное описание формата JSON-сообщений см. [здесь](#).

В рамках данного раздела, все команды будут возвращать результат своей работы в текстовом формате. В качестве примера возврата в формате JSON-сообщений будет рассмотрен вывод команды `help`. Также следует понимать, что в формате JSON-сообщений будет выполняться возврат абсолютно всех команд агента, включая и команды получения справочной информации.

Следует помнить, что все примеры, приведенные в данном разделе, не являются законченными конструкциями и предназначены только для демонстрации работы того или иного механизма или приема использования.

### 4.7.3. Команда help

Данная команда предназначена для получения справочной информации. Команда `help` без параметров покажет общую информацию по использованию режима агента.

[Копировать в буфер обмена](#)

```
designer> help
Использование:
    help [options] [arguments]
Общие параметры:
    --version | -v
        получение версии утилиты
Режим:
    help (h)
        Отображение справочной информации для указанного режима.
Аргументы:
    MODE
        режим, для которого необходимо получить информацию о ключах командной строки
Поддерживаемые режимы:
help          Отображение справочной информации для указанного режима.
common        Общие команды
options       Управление настройками
config        Режим работы с конфигурацией
infobase-tools Сервисные функции работы с информационной базой
designer>
```

Если требуется получить более конкретную информацию о группе команд или какой-то конкретной команде, то следует всю эту информацию ввести после ключевого слова `help`. Например, для получения информации о группе команд `options` необходимо выполнить команду `help options`.

[Копировать в буфер обмена](#)

```
designer> help options
Использование:
    options [command] [options]
Общие параметры:
    --version | -v
        получение версии утилиты
Режим:
    options
        Управление настройками
Команды:
    list
        Вывести значения всех настроек
    get
        Получить значение настройки
        --output-format
            Формат вывода
        --show-prompt
            Показывать приглашение командной строки
    set
        Установить значение настройки
        --output-format=<text|json>
            Формат вывода
        --show-prompt=<yes|no>
            Показывать приглашение командной строки
designer>
```

Для получения версии системы «1С:Предприятие», которая выполняется в режиме агента, необходимо выполнить команду `help --version`.

[Копировать в буфер обмена](#)

```
designer> help --version
8.3.10.2000
designer>
```

Если установить формат вывода в JSON-сообщения, то вывод информации о версии платформы «1С:Предприятие» будет выглядеть следующим образом:

[Копировать в буфер обмена](#)

```
designer> options set --output-format=json
[
{
"type": "success",
"message": ""
}
]
designer> help --version
[
{
"type": "success",
"body": "8.3.10.1772"
}
]
designer>
```

В данном примере первая команда (`options set ...`) устанавливает формат вывода, а следующая команда получает версию системы «1С:Предприятие».

#### 4.7.4. Команды группы `common`

Команды группы `common` отвечают за общие операции. В состав группы входят следующие команды:

- `connect-ib` - выполнить подключение к информационной базе, параметры которой указаны при старте режима агента.
- `disconnect-ib` - выполнить отключение от информационной базы, подключение к которой ранее выполнялось с помощью команды `connect-ib`.
- `shutdown` - завершить работу конфигуратора в режиме агента.

Команды данной группы не имеют параметров.

#### 4.7.5. Команды группы `options`

##### 4.7.5.1. Назначение группы команд

Команды группы `options` отвечают за управление настройкам текущей сессии. В состав группы входят следующие команды:

##### 4.7.5.2. `list`

С помощью данной команды можно просмотреть список параметров и их (параметров) текущее состояние.

##### 4.7.5.3. `get`

Данная команда позволяет получить значения параметров. Для команды доступны следующие параметры:

- `--output-format` - позволяет указать формат вывода результата работы команд:
  - `text` - команды возвращают результат в текстовом формате.
  - `json` - команды возвращают результат в формате JSON-сообщений.
- `--show-prompt` - позволяет управлять наличием приглашения командной строки `designer>`:
  - `yes` - в командной строке есть приглашение;
  - `no` - в командной строке нет приглашения.



- `--notify-progress` - позволяет получить информацию об отображении прогресса выполнения команды.
- `--notify-progress-interval` - позволяет получить интервал времени, через который обновляется информация о прогрессе.

#### 4.7.5.4. set

Данная команда позволяет выполнить установку значений параметров. Для команды доступны следующие параметры:

- `--output-format` - позволяет указать формат вывода результата работы команд:
  - `text` - команды возвращают результат в текстовом формате.
  - `json` - команды возвращают результат в формате JSON-сообщений.
- `--show-prompt` - позволяет управлять наличием приглашения командной строки `designer>`:
  - `yes` - в командной строке есть приглашение;
  - `no` - в командной строке нет приглашения.
- `--notify-progress` - позволяет управлять выводом информации о прогрессе выполнения команды:
  - `yes` - отображается прогресс;
  - `no` - прогресс не отображается (значение по умолчанию).

Список команд, поддерживающих отображение прогресса см. [здесь](#).

- `--notify-progress-interval` - позволяет указать интервал времени, через который обновляется информация о прогрессе. Значение задается с точностью 0.1 секунды. Значение по умолчанию равно 1 секунда. Нарушение точности приводит к формированию ошибки `CommandLineFormaError`. Список команд, поддерживающих отображение прогресса см. [здесь](#).

### 4.7.6. Команды группы config

#### 4.7.6.1. Назначение группы команд

Команды группы `config` отвечают за команды редактирования конфигурации.

#### 4.7.6.2. dump-config-to-files

Команда позволяет выполнить выгрузку конфигурации в xml-файлы. Для команды доступны следующие параметры:

- `--dir <путь>` - содержит путь к каталогу, в который будет выгружена конфигурация. Параметр является обязательным.
- `--extension <имя расширения>` - содержит имя расширения, которое будет выгружено в файлы.
- `--all-extensions` - если параметр указан, то в файлы будут выгружены все расширения конфигурации.
- `--format [hierarchical|plain]` - определяет формат выгрузки. По умолчанию используется иерархический формат выгрузки (`hierarchical`).
- `--update` - обновить существующую выгрузку. В этом случае будут выгружены только те объекты, версии которых отличаются от версий, указанных в файле `ConfigDumpInfo.xml`.
- `--force` - выполнить полную выгрузку, если при попытке обновления выгрузки (параметр `update`) выяснилось, что текущая версия формата выгрузки не совпадает с версией формата выгрузки, которая указана в файле `ConfigDumpInfo.xml`.

- `--get-changes <путь>` - сформировать файл, который содержит изменения между текущей и указанной выгрузками конфигурации.
- `--config-dump-info-for-changes <путь>` - путь к файлу [ConfigDumpInfo.xml](#), который используется для формирования файла изменений между двумя выгрузками конфигурации.
- `--list-file <файл>` - выгрузить только объекты метаданных и/или внешние свойства, указанные в файле, вне зависимости от того были они изменены или нет.

#### 4.7.6.3. load-config-from-files

Команда позволяет выполнить загрузку конфигурации из xml-файлов. Для команды доступны следующие параметры:

- `--dir <путь>` - содержит путь к каталогу, из которого будет загружена конфигурация. Параметр является обязательным.
- `--extension <имя расширения>` - содержит имя расширения, которое будет загружено из файлов.
- `--all-extensions` - если параметр указан, то из файлов будут загружены все расширения конфигурации.
- `--format [hierarchical|plain]` - определяет формат выгрузки. По умолчанию используется иерархический формат выгрузки (`hierarchical`).
- `--files <файл[, файл]>` - список файлов, которые требуется загрузить. Файла разделяются запятыми. Пути к файлам указываются относительно каталога загрузки. Абсолютные пути не поддерживаются. При использовании параметра `--list-file`, данный параметр не используется.
- `--list-file <файл>` - путь к файлу, в котором перечислены загружаемые файлы. Одна строка соответствует одному файлу. Пути к файлам указываются относительно каталога загрузки. Абсолютные пути не поддерживаются. При использовании параметра `--files`, данный параметр не используется.
- `--update-config-dump-info` - после окончания загрузки создать в директории файл [ConfigDumpInfo.xml](#), соответствующий загруженной конфигурации.

#### 4.7.6.4. dump-cfg

Команда позволяет выполнить выгрузку конфигурации или расширения в файл. Для команды доступны следующие параметры:

- `--file <путь>` - путь к файлу конфигурации (cf-файл) или расширению (cfe-файл).
- `--extension <имя расширения>` - содержит имя расширения, которое будет выгружено в файл.

#### 4.7.6.5. load-cfg

Команда позволяет выполнить загрузку конфигурации или расширения из файла. Для команды доступны следующие параметры:

- `--file <путь>` - путь к файлу конфигурации (cf-файл) или расширению (cfe-файл).
- `--extension <имя расширения>` - содержит имя расширения, которое будет загружено из файла.

#### 4.7.6.6. dump-external-data-processor-or-report-to-files

Команда позволяет выполнить выгрузку внешних обработок или отчетов в xml-файлы. Для команды доступны следующие параметры:

- `--file <файл>` - содержит имя файла, который будет выступать в роли корневого файла выгрузки внешней обработки/отчета в формате XML. Параметр является обязательным.
- `--ext-file <файл>` - полное имя файла с выгружаемой внешней обработкой (`*.epf`) или отчетом (`*.erf`).
- `--format [hierarchical|plain]` - определяет формат выгрузки. По умолчанию используется иерархический формат выгрузки (`hierarchical`).

#### 4.7.6.7. load-external-data-processor-or-report-from-files

Команда позволяет выполнить загрузку внешних обработок или отчетов из xml-файлов. Для команды доступны следующие параметры:

- `--file <файл>` - содержит имя файла, который является корневым файлом выгрузки внешней обработки/отчета в формате XML. Параметр является обязательным.
- `--ext-file <файл>` - полное имя файла с загружаемой внешней обработкой (`*.epf`) или отчетом (`*.erf`).

#### 4.7.6.8. update-db-cfg

Команда позволяет выполнить обновление конфигурации базы данных. Для команды доступны следующие параметры:

- `--prompt-confirmation` - определяет необходимость запроса у пользователя подтверждения о принятии изменений при реструктуризации информационной базы.
- `--dynamic-enable` - сначала выполняется попытка динамического обновления, если она завершена неудачно, будет запущено фоновое обновление.
- `--dynamic-disable` - указание данного параметра запрещает динамическое обновление.
- `--warnings-as-errors` - при указании данного параметра все предупреждения, которые могут возникнуть при обновлении конфигурации базы данных, будут считаться ошибками.
- `--background-start` - при указании данного параметра будет запущено фоновое обновление конфигурации, а текущий сеанс будет завершен.
- `--background-cancel` - при указании данного параметра происходит отмена запущенного фонового обновления конфигурации базы данных.
- `--background-finish` - при указании данного параметра запущенное фоновое обновление конфигурации базы данных будет завершено. При этом на базу данных будет наложена монопольная блокировка и проведена финальная фаза обновления.
- `--background-resume` - при указании данного параметра система продолжает фоновое обновление конфигурации базы данных, приостановленное ранее.
- `--server` - данный параметр указывает, что обновление конфигурации базы данных необходимо выполнить на стороне сервер «1С:Предприятия».
- `--extension <имя расширения>` - имя расширения.

При выполнении команды `update-db-cfg` используется следующий алгоритм работы:

- В том случае, если невозможно монопольно заблокировать базу данных и динамическое обновление возможно (не указан параметр `--dynamic-disable`) - будет выполнено динамическое обновление конфигурации базы данных.
- В том случае, если монопольно заблокировать базу данных возможно, но не требуется выполнять реструктуризацию базы данных - будет выполнено обычное обновление конфигурации базы данных.
- В том случае, если монопольно заблокировать базу данных возможно и требуется выполнить реструктуризацию базы данных, то будет выполнена следующая последовательность действий:

- выполняется определение списка измененных объектов;
- полученный список отображается в консоли;
- если указана необходимость подтверждения пользователя на принятие изменений (параметр `--prompt-confirmation`) - пользователю задается соответствующий вопрос;
- если ответ на вопрос отрицательный - обновление отменяется с выдачей соответствующего уведомления;
- если ответ утвердительный или запрос не требовался - обновление продолжается штатным образом.

#### 4.7.6.9. mobile-app-write-file

Команда позволяет сохранить конфигурацию для создания мобильного приложения в xml-файл. Для команды доступны следующие параметры:

- `--file <имя файла>` - определяет имя файла, в который будет сохранена конфигурация.

#### 4.7.6.10. mobile-client-write-file

Команда позволяет сохранить конфигурацию для создания мобильного клиента в xml-файл. Для команды доступны следующие параметры:

- `--file <имя файла>` - определяет имя файла, в котором расположен закрытый ключ, который используется для формирования подписи.

#### 4.7.6.11. mobile-client-digi-sign

Команда позволяет сформировать цифровую подпись конфигурации мобильного клиента. Для команды доступны следующие параметры:

- `--file <имя файла>` - определяет имя файла, в котором расположен закрытый ключ, который используется для формирования подписи.

#### 4.7.6.12. manage-cfg-support

Команда позволяет снимать конфигурация с поддержки. Допустимо использование следующих параметров:

- `--disable-support` - указывает на необходимость снятия конфигурации с поддержки. При отсутствии параметра генерируется ошибка.
- `--force` - выполнить снятие конфигурации с поддержки даже в том случае, если в конфигурации запрещены изменения. При отсутствии параметра будет сгенерирована ошибка, если попытка снятия с поддержки будет выполняться для конфигурации, для которой в интерактивном режиме управления поддержкой запрещены изменения.

#### 4.7.6.13. Группа команд extensions

##### 4.7.6.13.1. Назначение группы команд

Группа команд `config extensions` предназначена для управления расширениями с помощью режима агента.

##### 4.7.6.13.2. create

Команда предназначена для создания расширения в информационной базе. Расширение создается пустым. Для загрузки расширения следует использовать команду `config load-cfg` или `config load-config-from-files`. Допустимо использование следующих параметров:

- `--extension <имя>` - задает имя расширения. Параметр является обязательным.

- `--name-prefix <префикс>` - задает префикс имени для расширения. Параметр является обязательным.
- `--synonym <синоним>` - синоним имени расширения. Многоязычная строка в формате функции `Nstr()`.
- `--purpose <назначение>` - назначение расширения. `<Назначение>` может принимать следующие значения:
  - `customization` - назначение *Адаптация* (значение по умолчанию);
  - `add-on` - назначение *Дополнение*;
  - `patch` - назначение *Исправление*.

#### 4.7.6.13.3. delete

Команда предназначена для удаления расширения из информационной базы. Допустимо использование следующих параметров:

- `--extension <имя>` - задает имя удаляемого расширения.
- `--all-extensions` - указывает, что необходимо удалить все расширения.

#### 4.7.6.13.4. Группа команд properties

##### Назначение группы команд

Группа команд `config extensions properties` позволяет задавать и получать свойства расширения.

##### get

Команда предназначена для получения свойств расширения, расположенного в информационной базе. Допустимо использование следующих параметров:

- `--extension <имя>` - задает имя расширения, для которого необходимо получить свойства.
- `--all-extensions` - указывает, что необходимо получить свойства всех расширений, загруженных в информационную базу.

##### set

Команда предназначена для установки свойств расширения, расположенного в информационной базе. Допустимо использование следующих параметров:

- `--extension <имя>` - задает имя расширения, для которого необходимо установить свойства.
- `--active <режим>` - определяет активность расширения. `<Режим>` может принимать следующие значения:
  - `yes` - расширение активно.
  - `no` - расширение не активно.
- `--safe-mode <режим>` - определяет работу в безопасном режиме. `<Режим>` может принимать следующие значения:
  - `yes` - расширение работает в безопасном режиме.

- `no` - расширение работает в небезопасном режиме. В этом случае имя профиля безопасности автоматически сбрасывается (имя профиля устанавливается равным пустой строке).
- `--security-profile-name <профиль>` - определяет имя профиля безопасности, под управлением которого работает расширение. Если задается имя профиля безопасности, то автоматически устанавливается и признак работы в безопасном режиме.
- `--unsafe-action-protection <режим>` - определяет режим защиты от опасных действий. `<Режим>` может принимать следующие значения:
  - `yes` - защита от опасных действий в расширении включена.
  - `no` - защита от опасных действий в расширении отключена.
- `--used-in-distributed-infobase <режим>` - определяет возможность работы расширения в распределенной информационной базе. `<Режим>` может принимать следующие значения:
  - `yes` - расширение используется в распределенной информационной базе.
  - `no` - расширение не используется в распределенной информационной базе.
- `--scope <область действия>` - область действия расширения. `<Область действия>` может принимать следующие значения:
  - `infobase` - расширение действительно для все информационной базы.
  - `data-separation` - расширение действительно для области данных.

#### 4.7.6.14. sign-cfg

Команда позволяет подписать расширение конфигурации цифровой подписью. После выполнения процедуры подписи расширение может быть использовано в базовых версиях конфигураций, подписанных тем же ключом, что и расширение. Допустимо использование следующих параметров:

- `--result-file <путь к cfe-файлу>` - путь к результирующему файлу.
- `--type <тип конфигурации>` - местоположение подписываемого расширения конфигурации. Параметр может принимать следующие значения:
  - `extension-configuration` - расширение конфигурации.
  - `extension-db-configuration` - расширение конфигурации, расположенное в базе данных.
  - `extension-configuration-repository` - расширение конфигурации, расположенное в хранилище расширения конфигурации.
  - `file` - файл расширения конфигурации.
- `--name <имя расширения>` - имя расширения. Используется, если типа конфигурации принимает одно из следующих значений: `extension-configuration`, `extension-db-configuration`, `extension-configuration-repository`.
- `--version <версия >` - версия расширения конфигурации в хранилище. Используется при указании типа конфигурации `extension-configuration-repository`.
- `--file <путь к cfe-файлу>` - путь к файлу с подписываемым расширением. Используется при указании типа конфигурации `file`.
- `--digisign <имя файла с параметрами лицензирования>` - позволяет указать параметры лицензирования рабочего места пользователя.

### 4.7.7. Команды группы infobase-tools

#### 4.7.7.1. Назначение группы команд



Команды группы `infobase-tools` отвечают за получение сервисной информации об информационной базе.

#### 4.7.7.2. debug-info

С помощью данной команды получить информацию о настройках отладчика для информационной базы. Для получения доступны следующие параметры настроек отладчика:

- `enabled` - признак включения отладки.
- `protocol` - протокол отладки: `tcp` или `http`.
- `server-address` - адрес сервера отладки для данной информационной базы.

#### 4.7.7.3. data-separation-common-attribute-list

Данная команда позволяет получить список имен разделителей информационной базы.

#### 4.7.7.4. dump-ib

Команда предназначена для выполнения выгрузки информационной базы в dt-файл. Допустимо использование следующих параметров:

- `--file <имя файла>` - определяет имя dt-файла.

#### 4.7.7.5. restore-ib

Команда предназначена для выполнения загрузки информационной базы из dt-файл. Допустимо использование следующих параметров:

- `--file <имя файла>` - определяет имя dt-файла.

#### 4.7.7.6. erase-data

Команда выполняет удаление данных информационной базы.

### 4.7.8. Формат JSON-сообщений

В том случае, если установлен формат вывода в виде JSON-сообщений, сервер всегда будет возвращать массив из различной комбинации следующих значений:

- Имя: `type`, обязательное.
  - Возможные значения: `log` | `success` | `error` | `canceled` | `question` | `dbstru` | `loading-issue` | `progress` | `extension-info`.
  - Описывает тип сообщения:
    - `log` - информационное сообщение.
    - `success` - операция успешно завершена.
    - `error` - операция завершена с ошибкой.
    - `canceled` - операция отменена.
    - `question` - вопрос пользователю.
    - `dbstru` - информация о процессе реструктуризации.
    - `loading-issue` - ошибки и предупреждения, накопленные за время загрузки конфигурации из файлов.
    - `progress` - информация о прогрессе выполнения команды.

- `extension-info` - информация о расширении, которое находится в информационной базе.
- Имя: `error-type`
  - Возможные значения: отсутствуют.
  - Содержит тип ошибки, которая возникла при выполнении команды:
    - `UnknownError` - неизвестная ошибка.
    - `DesignerNotConnectedToInfoBase` - соединение с информационной базой не установлено.
    - `DesignerAlreadyConnectedToInfoBase` - соединение с информационной базой уже установлено.
    - `CommandFormatError` - неверный формат команды.
    - `DBRestructInfo` - ошибка реструктуризации базы данных.
    - `InfoBaseNotFound` - информационная база не найдена.
    - `AdministrationAccessRightRequired` - для выполнения операции требуются административные права.
    - `ConfigFilesError` - ошибки в процессе загрузки/выгрузки конфигурации из/в файла.
    - `DesignerAlreadyStarted` - обнаружен запущенный конфигуратор.
    - `InfoBaseExclusiveLockRequired` - требуется исключительная блокировка информационной базы.
    - `LanguageNotFound` - язык не обнаружен.
    - `ExtensionWithDataIsActive` - расширение конфигурации активно и содержит данные.
    - `ExtensionNotFound` - расширение не обнаружено.
- Имя: `message`
  - Возможные значения: отсутствуют.
  - Содержит локализованное сообщение об ошибке.
- Имя: `body`
  - Возможные значения: JSON-сообщение.
  - Содержание данного значения зависит от конкретной операции.

#### 4.7.9. Оповещение о прогрессе выполнения длительных операций

Некоторые команды могут требовать для своего исполнения существенного времени. Имеется возможность настроить режим агента таким образом, что прогресс выполнения команды будет отображаться или в виде сообщений или непосредственно на экране консоли. Настройка отображения прогресса выполняется с помощью команды `options set` (подробнее см. [здесь](#)).

Команды оповещают информацию о прогрессе, только в том случае, если в настройках включен соответствующий режим. Частота обновления информации также устанавливается параметром настроек. Прогресс измеряется в процентах. Если за интервал времени прогресс не изменился, то сообщение не отправляется.

Отображение прогресса поддерживают следующие команды:

- `dump-config-to-files;`
- `load-config-from-files;`

- `dump-external-data-processor-or-report-to-files;`
- `load-external-data-processor-or-report-from-files;`
- `dump-cfg;`
- `load-cfg;`
- `update-db-cfg.`

#### 4.7.10. Рекомендации по настройке SSH-клиентов

При настройке SSH-клиента PuTTY рекомендуется выполнить следующие настройки:

- Группа параметров **Terminal**:
  - параметр **Local echo - Force on**;
  - параметр **Local line ending - Force on**.
- Группа параметров **Window - Translation**:
  - параметр **Remote character set - UTF-8**.
- Группа параметров **Connection - SSH - TTY**:
  - включить параметр **Don't allocate a pseudo-terminal**.

### 4.8. Агент клиентского приложения 1cesla

#### 4.8.1. Общая информация

Приложение **1С:Предприятие - оповещение и запуск** (далее «агент клиентского приложения») предназначено для получения сообщений от сервера системы взаимодействия, а также для централизованного отображения оповещений. При этом сообщения получают и отображаются также и в том случае, если само приложение (для которого предназначено сообщение) в данный момент не запущено. Агент клиентского приложения, кроме накопления и отображения оповещений, позволяет быстро переходить к клиентскому приложению, для которого поступило сообщение, а также управлять отображением оповещений. Если приложение не запущено, агент выполняет запуск этого приложения. Также агент клиентского приложения, будучи запущенным, заменяет все пиктограммы информационных баз, которые (в ОС Windows) находятся в системном лотке панели задач, на единственную пиктограмму агента.

Агент клиентского приложения поставляется в следующих вариантах:

- исполняемый файл для ОС Windows в 32-разрядном варианте;
- исполняемый файл для ОС Linux в 32- и 64-разрядном вариантах;
- расширение веб-браузера для Google Chrome, Mozilla Firefox в 32- и 64-разрядном вариантах (в зависимости от разрядности используемого веб-браузера), а также расширение Microsoft Internet Explorer в 32-разрядном варианте.

#### 4.8.2. Установка агента

При установке «1С:Предприятия» (кроме отдельного дистрибутива тонкого клиента) дистрибутив агента клиентского приложения размещается в:

- Для ОС Windows: каталоге **ExtDst** конкретной версии;
- Для ОС Linux:
  - 32-разрядная версия системы: `/opt/1cv8/i386/A.B.C.D/ExtDst`;
  - 64-разрядная версия системы: `/opt/1cv8/x86_64/A.B.C.D/ExtDst`.

Файл пакета агента имеет имя вида **1c-enterprise-client-application-agent-a.b.c.d.arch**, где:

- **a.b.c.d** - это полный номер версии агента, расположенного в пакете, который соответствует номеру в виде **a.b.c.d**.
- **arch** - архитектура утилиты. Может принимать следующие значения:
  - OC Windows:
    - **x86** - 32-разрядная версия агента.
  - OC Linux:
    - **i386** - 32-разрядная версия агента;
    - **x86\_64** - 64-разрядная версия агента.

Для OC Windows пакет имеет расширение **.exe**, для OC Linux - **.sh**.

Установка агента инициируется запуском соответствующего установочного пакета вручную, с помощью встроенного языка или из диалога настроек системы взаимодействия клиентского приложения. Агент клиентского приложения устанавливается в следующие каталоги:

- OC Windows:
  - 32-разрядная система в 64-разрядной ОС:  
**%PROGRAMFILES(x86)%\1C\1CE\1cecla\ecs\_ver\_<api-version>**.
  - В остальных случаях: **%PROGRAMFILES%\1C\1CE\1cecla\ecs\_ver\_<api-version>**.
- OC Linux:
  - **~/bin/1cecla/ecs\_ver\_<api-version>** (вне зависимости от разрядности ОС).
  - Выражение **<api-version>** представляет собой цифру, равную версии протокола взаимодействия с сервером системы взаимодействия.

Имя исполняемого файла агента клиентского приложения: **1cecla**.

На компьютере в каждый момент времени может быть установлена только одна версия агента клиентского приложения.

Если во время установки платформы «1С:Предприятие» обнаруживается, что агент клиентского приложения установлен, то выполняется определение версии установленного агента. В том случае, если версия установленного приложения агента меньше версии агента, который поставляется с устанавливаемой платформой, то выполняются следующие действия:

- устанавливается (обновляется) агент клиентского приложения до актуальной версии;
- обновляется расширение веб-браузера, если это необходимо;
- выполняется перезапуск агента клиентского приложения, если он был запущен перед установкой новой версии платформы «1С:Предприятие».

При запуске клиентского приложения проверяется установленная и запущенная версия агента. Если она меньше доступной версии агента, то пользователю выводится диалог с информацией о наличии более свежей версии агента и гиперссылкой на запуск обновления. Нажатие на гиперссылку приводит к запуску процесса обновления.

После установки агента клиентского приложения в ОС Windows, оно будет автоматически запускаться при старте системы. При работе под управлением ОС Linux, пользователю следует самостоятельно указать, что агент клиентского приложения должен запускаться при старте системы. При этом следует помнить, что агенту, к моменту запуска, необходим функционирующий графический интерфейс.

### 4.8.3. Программный интерфейс агента

Для доступа к программному интерфейсу агента клиентского приложения служит менеджер агента клиентского приложения, который доступен через свойство глобального контекста

**АгентКлиентскогоПриложения**.

С помощью программного интерфейса может выполнять следующие действия:

- выполнить установку агента;
- проверить наличие новых версий агента;
- получать/устанавливать представление клиентского приложения в контекстном меню агента;
- подключать/отключать текущее клиентское приложение от агента;
- подключать/отключать обработчик изменения состояния агента.

Подключение агента к запущенному клиентскому приложению выполняется автоматически, по факту запуска клиентского приложения. Однако если возникает необходимость, чтобы агент клиентского приложения отображал оповещения и в том случае, если клиентское приложение в данный момент не запущено, то следует воспользоваться методом `АгентКлиентскогоПриложения.НачатьПодключение()`. Фактическое состояние подключения можно получить с помощью обработчика, который подключается методом `АгентКлиентскогоПриложения.ОтключитьОбработчикИзмененияСостояния()`. Для изменения представления приложения в списке агента следует использовать методы `АгентКлиентскогоПриложения.ПолучитьНаименованиеПриложения()` / `АгентКлиентскогоПриложения.УстановитьНаименованиеПриложения()`.