

# Система лицензирования и защиты конфигураций платформы 1С:Предприятие 8, версия 3.0

## Отличия от предыдущих версий

Подробнее пример использования СЛК 3.0 см. демонстрационную конфигурацию [LicenceDemoDb.cf](#) из комплекта поставки.

## Подключение компоненты и создание объекта

Компонента СЛК экспортирует два класса [AddIn.Licence.LicenceExtension20](#) и [AddIn.Licence.LicenceExtension](#). Класс [AddIn.Licence.LicenceExtension20](#) предназначен для совместимости с СЛК 2.1, для использования всех возможностей СЛК 3.0 необходимо использовать тип [AddIn.Licence.LicenceExtension](#):

```
Процедура ПодключитьКомпоненту() Экспорт
    // Подключение компоненты из общего макета, в который должен быть
    // загружен zip-архив с модулями компоненты для всех поддерживаемых
    // операционных систем.
    //
    // Файл архива с компонентой в поставке имеет вид licenceaddin-{%version%}-template.zip

    // Обязательно указываем тип компоненты ТипВнешнейКомпоненты.Native
    Если НЕ ПодключитьВнешнююКомпоненту("ОбщийМакет.КомпонентаСЛК", "Licence",
    ТипВнешнейКомпоненты.Native) Тогда
        СисИнфо = Новый СистемнаяИнформация;
        ВызватьИсключение "Ошибка подключения компоненты СЛК " + СисИнфо.ТипПлатформы;
    КонецЕсли;
КонецПроцедуры

Функция ОбъектКомпоненты() Экспорт
    // Подключаем компоненту только в случае необходимости (см. блок Исключение)
    Попытка
        // Считаем, что компонента уже подключена
        Возврат Новый("AddIn.Licence.LicenceExtension");
    Исключение
        // Исключение возможно только если компонента еще не подключена
        ПодключитьКомпоненту();
        Возврат Новый("AddIn.Licence.LicenceExtension");
    КонецПопытки;
КонецФункции
```

## Создание менеджера объектов

Для отключения сообщений безопасности о загрузке обработок из внешних источников (реализовано в 1С:Предприятии 8.3.9.2033) необходимо при создании и настройке менеджера объектов использовать метод [УстановитьОбъектКомпоненты](#), в который передать созданный объект [ОписаниеЗащитыОтОпасныхДействий](#). Кроме этого в блоке обработки исключения рекомендуется вызвать метод объекта компоненты [ОшибкаСозданияОбъекта](#) для типизации и логгирования ошибки:

```
Функция ОбъектКомпоненты() Экспорт
    // Получение объекта компоненты
    Объект = ОбъектКомпоненты();

    // Установка параметров связи с сервером СЛК
    Объект.ПараметрыСвязи = ПолучитьПараметрыСвязи();
```

```

Если НЕ Объект.Запуск(СЛК.Серия()) Тогда
    ВызватьИсключение Объект.ПолучитьОписаниеОшибки();
КонецЕсли;

// Получаем имя менеджера объектов
Имя = Объект.ПолучитьМенеджерОбъектов();
Если Имя = Неопределено Тогда
    ВызватьИсключение Объект.ПолучитьОписаниеОшибки();
КонецЕсли;

// Отключаем предупреждения о загрузке обработок из непроверенных источников
Защита = Неопределено;
Попытка
    Защита = Новый ("ОписаниеЗащитыОтОпасныхДействий");
    Защита.ПредупреждатьОбОпасныхДействиях = Ложь;
Исключение
    // Исключение возможно на предыдущих версиях платформы без механизма
    // защиты от опасных действий
КонецПопытки;

// Создание менеджера объектов
Попытка
    Если Защита = Неопределено Тогда
        МенеджерОбъектов = ВнешниеОбработки.Создать(Имя, БезопасныйРежим());
    Иначе
        МенеджерОбъектов = ВнешниеОбработки.Создать(
            ВнешниеОбработки.Подключить(ПоместитьВоВременноеХранилище(Новый ДвоичныеДанные(Имя)), ,
            БезопасныйРежим(), Защита));
    КонецЕсли;
// Настраиваем менеджер объектов
МенеджерОбъектов.УстановитьОбъектКомпоненты(Объект, Защита);
Исключение
    // Логгируем и получаем описание ошибки
    ВызватьИсключение Объект.ОшибкаСозданияОбъекта(ОписаниеОшибки());
КонецПопытки;
Возврат МенеджерОбъектов;
КонецФункции

```

## Логгирование

В СЛК 3.0 реализовано логгирование основных событий и ошибок. С целью упрощения поиска файлов логов реализовано их получение средствами компоненты СЛК при помощи метода [ПолучитьЛог](#):

```

// Получение серверного лога в виде двоичных данных
Лог = СЛК.МенеджерОбъектов().ПолучитьЛог();
Лог.Записать("D:\licenceserver.log");

// Получение лога компоненты в виде строки
Лог = СЛК.МенеджерОбъектов().ПолучитьЛог(Ложь, Истина);

```

Метод реализован как у менеджера объектов, так и объекта компоненты, что позволяет получить лог компоненты до создания менеджера объектов (например, в случае невозможности соединиться с сервером).

## Идентификация ошибок

Сообщения об ошибках начинаются с имени класса исключения в круглых скобках, например:

(EVersionMismatch) Устаревшая версия СЛК 3.0.8.5552, для работы конфигурации необходима СЛК 3.0.10 или выше

Разработчик конфигурации может использовать класс исключения для реализации дополнительной логики при обработке ошибок, например, при ошибках связи открывать окно настроек параметров и т.п.

Пример извлечения класса исключения см. в функции [ПолучитьКлассИсключения](#) общего модуля [СЛКОбщие](#) в демонстрационной конфигурации.

Полный список классов исключений см. Использование.

## Получение информации о сервере

В СЛК 3.0 реализована возможность получения средствами компоненты подробной информации о сервере СЛК при помощи методов менеджера объектов [ПолучитьИнформациюОСервере](#) / [ПопыткаПолучитьИнформациюОСервере](#), которые возвращают фиксированную структуру со следующими полями:

|                 |  |
|-----------------|--|
| ComputerName    | Имя компьютера, на котором установлен сервер СЛК |
| LocalIP         | IP адрес   |
| ComputerID      | ID компьютера                                    |
| OSType          | Тип ОС   |
| OSVersion       | Версия ОС  |
| ModuleName      | Путь к исполняемому модулю сервера               |
| PID             | ID процесса                                      |
| CPU             | Тип (Разрядность) процесса                       |
| Version         | Версия   |
| VersionOutdated | Признак, что сервер можно обновить               |
| ConsoleUrl      | Адрес веб консоли                                |

```
Менеджер = МенеджерОбъектов();
// Получение информации о сервере
Инфо = Менеджер.ПолучитьИнформациюОСервере();
Возврат Инфо.Version + " (" + Инфо.OSType + " " + Инфо.CPU + ", " + Инфо.ConsoleUrl + ") " +
    // Проверка возможности обновления
    ? (Инфо.VersionOutdated, " - рекомендуется обновить до версии " + Менеджер.Версия(), "");
```

## Отображение консоли сервера

Консоль сервера в СЛК 3.0 реализована в виде веб приложения, поэтому возможно показать ее непосредственно в конфигурации при помощи поля HTML документа. Адрес консоли возвращается при получении информации о сервере в поле [ConsoleUrl](#) (см.

выше). Пример см. в демонстрационной конфигурации в форме [КонсольСервера](#) обработки [СЛК](#).

## Установка лицензий непосредственно из конфигурации

С целью более полной интеграции с конфигурацией в СЛК 3.0 возможно выполнять установку программных лицензий непосредственно из конфигурации, как в автоматическом режиме, так и при помощи файлового или телефонного запроса. Для этого у менеджера объектов реализованы следующие методы:

[УстановитьЛицензию](#) / [ПопыткаУстановитьЛицензию](#)

[ОбновитьЛицензии](#) / [ПопыткаОбновитьЛицензии](#)

[СоздатьФайловыйЗапрос](#) / [ПопыткаСоздатьФайловыйЗапрос](#)

[УстановитьФайловыйОтвет](#) / [ПопыткаУстановитьФайловыйОтвет](#)

[СоздатьТелефонныйЗапрос](#) / [ПопыткаСоздатьТелефонныйЗапрос](#)

[УстановитьТелефонныйОтвет](#) / [ПопыткаУстановитьТелефонныйОтвет](#)

Пример использования см. в демонстрационной конфигурации в обработке [СЛК](#) в форме [УстановкаЛицензии](#).

## Получение лицензии

Вместо методов [ПроверитьЛицензиюСеанса](#) / [ПопыткаПроверитьЛицензиюСеанса](#) в СЛК 3.0 необходимо использовать методы [ПолучитьЛицензию](#) / [ПопыткаПолучитьЛицензию](#), которые возвращают фиксированную структуру параметров ключа защиты, лицензия которого была занята:

|                |   |
|----------------|---|
| Type           | Тип ( <a href="#">USB</a> для аппаратных, <a href="#">Virtual</a> для программных)                              |
| FileName       | Имя файла или символическая ссылка  |
| Enabled        | Признак доступности для получения лицензий  |
| IsBlank        | Признак неактивированного ключа (болванки)  |
| BlankKeyNo     | Серийный номер «болванки»   |
| Workable       | Признак работоспособности (исправности для аппаратных или соответствия конфигурации компьютера для программных) |
| OpenError      | Описание ошибки открытия  |
| Version        | Версия ключа  |
| KeyNo          | Серийный номер ключа  |
| KeyID          | Серия ключей  |
| KeyType        | Тип ключа   |
| LicenceCount   | Количество лицензий   |
| ActivationCode | Код активации, использовавшийся при активации ключа   |
| ActivationDate | Дата активации  |

|                 |                                  |
|-----------------|----------------------------------|
| Period          | Срок действия в днях             |
| ExpireDate      | Дата окончания действия          |
| DaysLeft        | Оставшиеся дни                   |
| Expired         | Признак, что срок действия истек |
| Flags*          | Флаги функциональности           |
| ProductVersion* | Версия продукта                  |
| StartDate*      | Дата начала действия             |
| ITN**           | ИНН                              |
| IEC**           | КПП                              |
| ProductName     | Наименование серии               |
| RegNo           | Регистрационный номер            |
| Article         | Артикул номенклатуры             |
| ArticleName     | Наименование номенклатуры        |

\* Устанавливаются при создании ключа в Центре Лицензирования

\*\* Устанавливаются при активации ключа у конечного пользователя

```
Лицензия = СЛК.ПолучитьЛицензию();
// Анализ номенклатуры (АртикулКорп и АртикулПРОФ - предопределенные константы)
Если Лицензия.Article = АртикулКорп Тогда
    // Разрешить функционал редакции КОРП
ИначеЕсли Лицензия.Article = АртикулПРОФ Тогда
    // Разрешить функционал редакции ПРОФ
КонецЕсли;
```

## Использование параметров контроля лицензии

В СЛК 3.0 реализована возможность указать дополнительные параметры получения лицензии (ИНН, КПП, Версия продукта, Рег. номер) при помощи метода менеджера объектов [УстановитьПараметрыКонтроляЛицензии](#). При этом лицензии будут заниматься только у ключей с соответствующими значениями параметров, остальные ключи будут игнорироваться.

Например, при указании ИНН/КПП конфигурация получит лицензию, только если на сервере СЛК установлены ключи с указанными значениями ИНН/КПП. Аналогично, при указании версии продукта, получение лицензии будет возможно, только если установлены ключи с соответствующим значением этого параметра.

Вызов этого метода должен выполняться в защищенном коде, например, в предопределенном методе защищенного объекта [ПриСозданииОбъекта](#).

```
Процедура ПриСозданииОбъекта(МенеджерОбъектов, Отказ, ОписаниеОшибки) Экспорт
    // Сохраняем менеджер объектов во внутренней переменной модуля
    _менеджерОбъектов = МенеджерОбъектов;

    // Пример контроля по ИНН/КПП - при указании этих параметров конфигурация
    // получит лицензию только если на сервере СЛК будут свободные ключ с
    // соответствующими значениями ИНН/КПП.
    _менеджерОбъектов.УстановитьПараметрыКонтроляЛицензии(, 77000000, 77000);
```

*КонецПроцедуры*

*Из соображений безопасности повторная установка параметров возможна только с теми же значениями, которые были использованы при первом вызове. В случае повторного вызова с другими значениями дальнейшая работа менеджера объектов блокируется.*

## Контроль только наличия ключа

В дополнении к возможностям СЛК 2.1 (указание значения **Истина** в методах получения лицензии и использование предопределенной процедуры защищенного объекта **ПередКонтролемЛицензии**) в СЛК 3.0 для контроля только наличия ключа защиты без учета количества лицензий необходимо использовать метод **УстановитьПараметрыКонтроляЛицензий**, передавая в первом параметре значение **Истина**:

```
// Контроль только наличие ключа
МенеджерОбъектов.УстановитьПараметрыКонтроляЛицензии (Истина) ;
```

После установки этого значения последующие вызовы методов **ПолучитьЛицензию** / **ПопыткаПолучитьЛицензию** не будут занимать лицензию.

*Из соображений безопасности такой вызов должен быть помещен в предопределенной процедуре **ПриСозданииОбъекта** всех защищенных объектов. В противном случае, при создании защищенного объекта режим контроля только наличия ключа будет отменен.*

## Получение списка установленных лицензий

Вместо методов **ПолучитьСписокКлючей** / **ПопыткаПолучитьСписокКлючей** в СЛК 3.0 необходимо использовать методы **ПолучитьЛицензии** / **ПопыткаПолучитьЛицензии**, которые возвращают фиксированный массив структур, аналогичных получаемым методами **ПолучитьЛицензию** / **ПопыткаПолучитьЛицензию** (см. выше).

При этом разработчик может получать список лицензий как своей серии, которой защищается конфигурация, так и всех, установленных на сервере СЛК.

При получении лицензий своей серии учитываются параметры контроля лицензии (см. ниже).

## Получение счетчиков

Для получения счетчиков занятых / свободных лицензий вместо методов [ПолучитьОбщееКоличествоЛицензий](#) / [ПопыткаПолучитьОбщееКоличествоЛицензий](#) в СЛК 3.0 необходимо использовать методы [ПолучитьСчетчики](#) / [ПопыткаПолучитьСчетчики](#), которые возвращают следующую структуру:

|                   |                                    |
|-------------------|------------------------------------|
| Licences          | Количество ключей                  |
| Connections       | Количество соединений              |
| Sessions          | Количество сеансов                 |
| TotalLicenceCount | Общее количество лицензий          |
| UseLicenceCount   | Количество использованных лицензий |
| FreeLicenceCount  | Количество свободных лицензий      |

При формировании значений учитываются параметры контроля лицензии (см. выше). Пример использования см. в демонстрационной конфигурации в обработке [СЛК](#) в форме [ПроверкаКоманд](#).

## Работа с пользовательской памятью и лицензионными параметрами

Для работы с пользовательской памятью ключей и лицензионными параметрами в СЛК 3.0 реализованы методы, в которых необходимо строго указывать С/Н ключа, с памятью которого предполагается работать или значение параметров для которого требуется получить:

[ПрочитатьПамять](#) / [ПопыткаПрочитатьПамять](#)

[ЗаписатьПамять](#) / [ПопыткаЗаписатьПамять](#)

[ОбнулитьПамять](#) / [ПопыткаОбнулитьПамять](#)

[УстановитьПароли](#) / [ПопыткаУстановитьПароли](#)

[ПолучитьПараметр](#) / [ПопыткаПолучитьПараметр](#)

Такой подход позволяет работать со всеми ключами, установленными на сервере СЛК, а не только с текущим. Пример использования см. в демонстрационной конфигурации в обработке [СЛК](#) в форме [ПроверкаКоманд](#).

## Создание файлов данных

*СЛК 3.0 совместима по форматам файлов данных с предыдущими версиями, поэтому для создания файлов данных возможно использовать редактор файлов данных от СЛК 2.1.*

Редактор файлов в СЛК 3.0 реализован в виде консольной программы, доступной для платформ Windows и Linux, для использования в автоматическом режиме. В отличие от

предыдущих версий редактор СЛК 3.0 может быть без ключа защиты – создание файлов данных выполняется при помощи открытого ключа разработчика, поставляемого в комплекте разработчика для конкретной серии ключей в виде текстового файла вида {Серия}.cryptkey.

#### Использование:

licenceedit <Команда> <Имя файла> [<Источник1>,<Источник2>...<ИсточникN>]  
[<Параметры>...]

#### Команды:

##### **c, co**

Создание файла защищенных объектов

##### **p, sp**

Создание файла лицензионных параметров

##### **l, list**

Вывод содержимого файла

#### Источник (и):

Имя или маска файлов исходных данных – файлов обработок / отчетов или конфигурационного файла лицензионных параметров.

*При создании файлов защищенных объектов, объектам присваиваются имена файлов без расширения. Например, обработка из файла «ЗащищеннаяОбработка.erf» получит имя «ЗащищеннаяОбработка».*

*В ОС **Windows** при использовании в именах файлов русских букв необходимо учитывать кодировку. Например, если редактор вызывается из командного файла в кодировке ansi (CP1251 для русских региональных настроек), то необходимо указать это при помощи ключа **-cen(--cmdencoding)=ansi**.*

#### Параметры:

##### **-s, --serie={KeyID}**

Серия ключа (если не указана, то извлекается из имени создаваемого файла)

##### **-ck, --cryptkey={FileName}**

Имя файла ключа шифрования (если не указан, то принимается <KeyID>.cryptkey)

##### **-y, --yes**

Автоматически отвечать «Да» на все запросы

##### **-cen, --cmdencoding={oem|ansi|utf8}**

Кодировка параметров командной строки (только для ОС Windows; по умолчанию «oem», как при выполнении в командной строке)



**-en, --encoding={ansi|utf8|utf16}**

Кодировка исходного файла лицензионных параметров (по умолчанию «ansi», для совместимости с СЛК 2.1)

## Коды возврата:

- 0 : Нет ошибок
- 1 : Неизвестная / системная ошибка
- 2 : Операция отменена пользователем
- 3 : Серия ключей не указана или указана с ошибкой
- 4 : Не найдены исходные данные
- 5 : Ошибка при получении ключа шифрования

## Примеры использования

**licenceedit c my.datafile "D:/Samples/\*.epf" -y --serie=672B**

Создание файла данных серии 672B и добавление в него всех внешних обработок из папки D:\Samples\. Если файл уже существует, то его содержимое будет автоматически перезаписано (ключ **-y**).

**licenceedit l my.datafile --serie=672B**

Вывод содержимого файла данных my.datafile

**licenceedit p 672B.paramfile "D:/Samples/ParamFile.config" -y**

Создание файла параметров серии 672B на основе конфигурационного файла из папки D:\Samples. Т.к. серия не указана явно, то извлекается из имени создаваемого файла **672B.paramfile**.

## Размещение и установка файлов данных

*Файлы данных рекомендуется загружать в общий макет конфигурации, поэтому установочный пакет сервера не устанавливает и автоматически.*

Если по каким-то причинам разработчик хочет использовать файлы именно на сервере СЛК, то необходимо скопировать их в папку установки сервера.

*Расположение в ОС Windows*

`%ProgramFiles%\1C\licence\3.0\`

*Расположение в ОС Linux*

`/opt/1C/licence/3.0/`