

127 018, Москва, Сущевский Вал, 18  
Телефон: (495) 995 4820  
Факс: (495) 995 4820  
<http://www.CryptoPro.ru>  
E-mail: [info@CryptoPro.ru](mailto:info@CryptoPro.ru)



Средство  
Криптографической  
Защиты  
Информации

КриптоПро JCP

Версия 2.0 R2

Использование  
класса-загрузчика  
новой лицензии

ЖТЯИ.00091-02 93 01

Листов 7

---

**© ООО "Крипто-Про", 2000-2018. Все права защищены.**

Авторские права на средство криптографической защиты информации «КриптоПро JCP» версия 2.0 R2 и эксплуатационную документацию зарегистрированы в Российском агентстве по патентам и товарным знакам (Роспатент).

Документ входит в комплект поставки программного обеспечения СКЗИ «КриптоПро JCP» версия 2.0 R2, на него распространяются все условия лицензионного соглашения. Без специального письменного разрешения ООО "КРИПТО-ПРО" документ или его часть в электронном или печатном виде не могут быть скопированы и переданы третьим лицам с коммерческой целью.

## Оглавление

<u>1.Введение.....</u>	<u>4</u>
<u>2.Использование класса-загрузчика новой лицензии.....</u>	<u>5</u>

# 1. Введение

В данной документации приводится описание использования класса-загрузчика новой лицензии.

## 2. Использование класса-загрузчика новой лицензии

Для установки новой лицензии можно воспользоваться одним из способов описанных в «Руководстве администратора».

Для установки новой лицензии с сервера лицензий необходимо выполнить следующие операции:

1. Создать класс, расширяющий абстрактный класс `JCP.tools.LicenseLoader` и реализующий метод `getNewLicense()`.
2. Записать имя класса в реестр с помощью метода `setLoaderName(String class_name)` класса `JCP.tools.License`.

Теперь по истечении лицензии «КриптоПро JCP» версия 2.0 R2 получает из реестра имя класса-установщика, который затем обращается к серверу, который в свою очередь пытается получить новую лицензию от сервера лицензий. Если всё проходит успешно (не превышено максимальное число лицензий, выдаваемых за определенный срок и т.д.), то клиент получает новую лицензию.

Также установка может быть инициирована пользователем посредством вызова метода `setNewLicense()` класса `JCP.tools.License` при условии выполнения двух вышеуказанных действий. При обработке запроса на новую лицензию от клиента серверу может потребоваться текущая клиентская лицензия. Ее можно получить, вызвав конструктор класса `JCP.tools.License` без параметров:

```
License current_license = new License();
```

Дату окончания действия лицензии можно получить, вызвав метод `getEndDate()`. Запись лицензии в реестр осуществляет метод `store()`. Метод `verifyLicense()` получает тип лицензии.

По умолчанию «КриптоПро JCP» версия 2.0 R2 использует собственный класс-загрузчик, который не обращается к серверу, а предлагает пользователю ввести лицензию в обычном диалоговом окне.

Ниже приведен пример создания класса-загрузчика и его использования:

Создание класса:

```
public class TestLicenseLoader extends ru.CryptoPro.JCP.tools.LicenseLoader {

    public AbstractLicense getNewLicense() throws Exception {
        License new_lic;
        ...
        // transporting all required data to the server
        // getting the new license
        ...
        return new_lic;
    }
}
```

Установка лицензии "вручную" пользователем:

```
...
// some code
String path = "....."; // path to the class
```

```
String class_name = path + "TestLicenseLoader";
ru.CryptoPro.JCP.tools.License.setLoaderName(class_name);
try {
    License dummyLicense = new License(null,null,null);
    dummyLicense.setNewLicense();
}
catch (Exception e) {
    // catching exception
    ...
}
// another code
...
```

Сигнатуры public-конструкторов, полей и методов класса ru.CryptoPro.JCP.tools.License:

#### Конструкторы:

- public License() throws IOException; - конструктор, получающий текущую лицензию из реестра. Выбрасывает исключение в случае ошибки чтения.
- public License(String srcUserName,String srcCompanyName,String srcProductID); - конструктор по имени пользователя, названию компании и серийному номеру.

#### Типы лицензии:

- PERMANENT\_LICENSE - верная постоянная лицензия,
- CORRECT\_TEMP\_LICENSE - верная временная лицензия,
- NEED\_NOTIFY - временная лицензия истекает,
- RUN\_OUT\_OF\_TIME - лицензия истекла,
- INCORRECT\_ID\_FORM - форма серийного номера неверна,
- INCORRECT\_PRODUCT\_TYPE - неверный тип продукта,
- INCORRECT\_ID\_HASH - неверный серийный номер,
- INCORRECT\_CPU\_AMOUNT - превышено допустимое число ЦПУ,
- INCORRECT\_FIRST\_DATE - неверная дата первой установки,
- INCORRECT\_ID\_SERVER - неверный тип лицензии - не серверная лицензия,
- INCORRECT\_ID\_CRYPT - неверный тип лицензии - не для шифрования,
- INCORRECT\_LICENSE\_VERSION - неверная версия лицензии.

#### Поля:

- public static final String STR\_VALID\_LICENSE = "Valid license."; - строка "лицензия верна".
- public static final long NOTIFY\_TIME = 24 \* 60 \* 60 \* 1000; - время до истечения срока действия лицензии, при котором КриптоПро JCP запрашивает ее обновление.
- public static final int SERIAL\_PRODUCTID\_NUM = 20; - размер серийного номера с разделителями.

#### Методы:

- public long getAllowedAmount(); - возвращает допустимое число процессоров.
- public String getCompanyName(); - получает имя компании.
- public String getDefaultLoaderName(); - получает имя класса-загрузчика, используемого по умолчанию.
- public static String getDefaultUserName(); - получает имя пользователя по умолчанию.
- public long getEndDate(); - возвращает WRONG\_LICENSE, если лицензия неверна, PERMANENT\_LICENSE, если лицензия неограничена, срок окончания действия в миллисекундах иначе.
- public String getProductID(); - получает лицензионный номер продукта.
- public String getUsername(); - получает имя пользователя, на которого зарегистрирована лицензия.

ЖТЯИ.00091-02 93 01. КриптоПро JCP. Использование класса-загрузчика новой лицензии

- `public String getVersion();` - получает номер версии лицензии.
- `public boolean isWriteAvailable();` - проверяет наличие необходимых прав для записи лицензии.
- `public void setLoaderName(String name);` - устанавливает имя класса-загрузчика новой лицензии и записывает его в реестр.
- `public void setNewLicense() throws Exception;` - применение описано выше.
- `public void store() throws ConfigurationException;` - осуществляет запись лицензии в реестр. Выбрасывает исключение в случае ошибки записи.
- `public int verifyLicense();` - проверяет корректность лицензии и возвращает ее тип.