

## Правила версионирования исходного кода, скомпилированного проекта и библиотеки в соответствии с процессом разработки ПО в АО "ОНИИП".

---

примечание: за основу взят один из общеиспользуемых\* вариантов версионирования в мировой практике разработки ПО.

\*применительно к ПО для дистрибутивов на базе ядра Linux и ОС семейства Windows

### # Исходный код

---

Формат имени архива с исходным кодом: <appname>\_<year>.<month>.<day>.<build>~<desc>\_<os>.zip

#### Описание полей и значений:

<appname> - название СПО;

<year>, <month>, <day> - год, месяц и число сборки конкретной версии, соответственно;

<build> - номер сборки, первоначальное значение 0;

<desc> - этап разработки, первоначальное значение iter1.

этапы разработки (в возрастающем порядке):

iter1, iter2, iter3 и т.д. - итерации (этапы разработки сроком 2 недели);

pi1, pi2, pi3 и т.д. - предварительные испытания;

gi1, gi2, gi3 и т.д. - государственные испытания;

ti1, ti2, ti3 и т.д. - типовые испытания.

<os> - целевая ОС:

win - ОС семейства Microsoft Windows без конкретизации (с конкретизацией: winxp, win7 и т.д.)

linux - дистрибутив на базе ядра Linux без конкретизации (с конкретизацией: debian, msvs и т.д.)

all - кроссплатформенный исходный код

#### Пример использования:

Нерпа: r-357nm\_2016.12.08.1~gi2\_linux.zip

Буран: buran\_2015.05.21.3~iter3\_all.zip

#### Хранение в Лоцмане:

Архив r-357nm\_2016.12.08.1~gi2\_linux.zip следует поместить в архив с соответствующим УИЯД, например УИЯД.02380-01\_12\_01.zip

#### Дополнительная информация:

1. Если проект на основе фреймворка Qt - версия дублируется в переменной VERSION (главный pro файл)
2. Если проект без использования Qt - версия дублируется в файле VERSION (находится внутри архива с исходным кодом)

.

### # Скомпилированный проект

---

После компиляции версия "зашита в главный исполняемый файл скомпилированного проекта", имя исполняемого файла остается без изменений.

#### Как посмотреть версию скомпилированного проекта

Чтобы узнать версию исполняемого файла СПО существует несколько вариантов:

а) в консоли выполнить команду: <appname> --version

в) если это графическое приложение, то после его запуска, в правом верхнем углу главного окна, будет указана версия.

Щелкнув на эту надпись мышкой, появится окно с дополнительной информацией.

Формат строки версии: <appname>\_<year>.<month>.<day>.<build>~<desc>\_<arch>\_<os>

#### Описание полей и возможные значения:

<appname> - название СПО. Устанавливается разработчиком;  
<year>, <month>, <day> - год, месяц и число сборки конкретной версии, соответственно. Автоматически устанавливается сервером сборки в случае успешной сборки;  
<build> - номер сборки. Первоначальное значение 0. При переходе к следующему этапу испытаний, т.е при смене <desc> обнуляется. Автоматически инкрементируется сервером сборки в случае успешной сборки;  
<desc> - описание, служит для записи этапа разработки. Первоначальное значение iter1. Устанавливается вручную (запрашиваются сервером сборки):  
    этапы разработки (перечислено в возрастающем порядке):  
    iter1, iter2, iter3 и т.д. - итерации (этапы разработки сроком 2 недели);  
    pi1, pi2, pi3 и т.д. - предварительные испытания;  
    gi1, gi2, gi3 и т.д. - государственные испытания;  
    ti1, ti2, ti3 и т.д. - типовые испытания.  
<arch> - архитектура целевой платформы: i386 или amd64.  
<os> - целевая ОС:  
    win - ОС семейства Microsoft Windows без конкретизации, winxp - Windows XP SP3, win7 - Windows 7 и т.д.  
    linux - дистрибутив на базе ядра Linux без конкретизации, debian - Debian, msvs - MCBC и т.д.

#### Пример использования:

Название СПО: P-357NM, дата сборки: 2016.12.08, номер сборки: 1, этап испытаний: итерация 2, архитектура целевой платформы: i386, целевая ОС: MCBC.

Строка версии СПО: r-357nm\_2016.12.08.1~iter2\_i386\_msvs

%{color:white}.%

h3. # Библиотека

примечание: версия устанавливается разработчиком вручную, за исключением <build>.

```
<pre>
Формат строки версии: <libname>.so.<major>.<minor>.<patch>.<build>
</pre>
```

\*Описание полей и возможные значения:\*

```
<pre>
<libname> - название библиотеки. Устанавливается разработчиком;
<major>, <minor>, <patch> - мажор, минор и патч сборки конкретной версии, соответственно. Устанавливается разработчиком;
<build> - номер сборки. Первоначальное значение 0. -При переходе к следующему этапу испытаний, т.е при смене <desc> обнуляется.- Автоматически инкрементируется;
</pre>
```

\*Пример использования:\*

```
<pre>
Название: Сеанс "Масленка" ОЦ, мажор: 1, минор:0, патч: 0, номер сборки: 2
```

Строка версии библиотеки: MaslenkaOcSessionGui.so.1.0.0

libMaslenkaOcSessionGui\_.so.1.0.0

Чтобы узнать версию исполняемого файла СПО существует несколько вариантов:

а) открыть текстовый файл VERSION находящийся в той же директории, что и исполняемый файл СПО  
б) в консоли выполнить команду: <appname> --version (ОПИСАТЬ ЧТО БУДЕТ ВЫВЕДЕНО)  
в) если это графическое приложение то после его запуска, следует посмотреть в правый верхний угол главного окна (сборка <build> от <day>.<month>.<year>)  
</pre>

%{color:white}.%

\*\* это совокупность плагинов, исполняемых файлов, конф. файлов, справки и т. д.

\*\*\* запрашиваемые параметры зависят от версии сервера сборки