

## ***Gem #158: GPRinstall- Часть 1***

**Автор:** Pascal Obry—EDF R&D

Краткое содержание: GPRinstall - инструмент, относительно недавно разработанный AdaCore, с помощью которого можно устанавливать проект вне зависимости от того, является он стандартным проектом, проектом библиотек, проектом библиотек агрегатов или стандартным проектом агрегатов. С ним можно забыть как о ручном копировании артефактов, так и о необходимости обеспечения надежной процедуры установки для всех платформ.

### **Давайте начнём...**

Построить библиотеку с помощью GPRbuild довольно просто. GPRbuild обрабатывает несколько языков и несколько цепочек инструментов. В принципе, большинство проектов можно построить с помощью простой команды, такой как:

```
$ gprbuild -p libprj.gpr
```

Обратите внимание, что ключ «-p» указывает gprbuild создать любые отсутствующие каталоги obj, lib или exec. Та же опция применяется к gprinstall. Через несколько минут или часов, в зависимости от размера базы кода, мы получаем набор объектного кода и / или библиотек. Все идет нормально!

Наш следующий шаг - установить все это для использования другим проектом. То есть у нас есть библиотека, которую мы хотим сделать доступной для другого проекта. Другому проекту нужны файлы спецификаций Ada, которые будут «засохшими - withed», и библиотечный файл (статический или динамический) для ссылки. Затем мы начинаем писать набор правил make-файла:

```
install:
    mkdir /some/prefix/include/prj
    mkdir /some/prefix/lib/prj
    cp lib/*.ali /some/prefix/
```

Подождите ... У нас есть cp на Windows? Нужно ли устанавливать файл .ali только для чтения? Нужно ли копировать тела? О, и, конечно, нам нужно предоставить проект, который будет установлен в нужном месте, чтобы иметь возможность с ним, а это “with” в файле проекта, работать из другого проекта.

Похоже, впереди еще много работы ...Или нет. Фактически, GPRinstall позаботится обо всем. Да, это действительно обо всем позаботится.

В частности, GPRinstall будет:

- копировать правильные источники, учитывая текущие соглашения об именах и исключения
- скопирует спецификацию в нужное место
- копировать тела в нужное место, только при необходимости
- скопирует объектный код или библиотеку в нужное место
- автоматически сгенерировать файл проекта и поместить его в нужное место (где его будет искать gprbuild)

- записывать все установленные файлы и предоставлять простой способ удалить именно то, что было установлено, не больше и не меньше.
- обеспечить способ установки нескольких сборок (отладка / производство, наиболее распространены статические / общие библиотеки), выбираемых с помощью переменной проекта

По умолчанию правильное место "right place" - это место, где установлен компилятор. Конечно же, в GPRinstall есть много переключателей – в виде ключей, указывающих, где должны быть установлены артефакты компиляции. Итак, чтобы установить вышеупомянутый проект библиотеки, нам нужно просто запустить:

```
$ gprinstall -p libprj.gpr
```

Если у нас есть вторая сборка для общей библиотеки:

```
$ gprbuild -p -XLIBRARY_TYPE=relocatable libprj.gpr
```

тогда мы можем установить его с помощью:

```
$ gprinstall -p -XLIBRARY_TYPE=relocatable --build-name=shared libprj.gpr
```

Обратите внимание, что мы указали здесь имя сборки, которое является просто строкой, используемой для идентификации конкретной установки. Эта строка добавляется в качестве возможного значения в переменную BUILD сгенерированного проекта. При первом вызове gprinstall мы использовали строку имени сборки по умолчанию, которая является "default". Итак, теперь у нас есть что-то вроде этого в сгенерированном проекте:

```
library project Libprj is
  type BUILD_KIND is ("default", "shared");
  BUILD : BUILD_KIND := external("LIBPRJ_BUILD", "default");
  for Languages use ("Ada");
  case BUILD is
    when "shared" =>
      for Source_Dirs use ("../../include/libprj.shared");
      for Library_Dir use "../../lib/libprj.shared";
      for Library_Kind use "relocatable";
    when "default" =>
      for Source_Dirs use ("../../include/libprj");
      for Library_Dir use "../../lib/libprj";
      for Library_Kind use "static";
  end case;
  ...
```

И если мы решим фактически удалить все установки этой библиотеки, нам просто нужно выполнить:

```
$ gprinstall --uninstall libprj
```

Вот и все. Подождите, нет, на самом деле есть еще одна вещь ... теперь мы можем удалить весь беспорядок make-файла!

### Связанный со статьёй текст программы

Файлы примеров Ada Gems распространяются AdaCore и могут быть использованы или изменены для любых целей без ограничений.

## Об авторе (из общедоступных источников)

Автор Pascal Obry



69, rue Jean Longuet , 92290 Châtenay-Malabry, FRA

Инженер-программист и стремится сделать надежные коды. Pascal Obry также увлечен фотографией.

### *Навыки*

Qualité Logicielle

Langage Ada

### *Ссылки*

<http://www.obry.net>

<http://v2p.fr.eu.org>

EDF R&D является отделом исследований и разработок EDF Group.

*Last Updated: 11/24/2017*

*Posted on: 4/28/2014*

### Обсуждение...

## 3 Comments

- **Bruce R. Barkstrom**  
May 1st, 2014

Which of the Unicode normalization forms (NFD, NFKD, NFC, or NFKC – Unicode Standard, Version 6.2, section 2.12, pp. 47-48) does the unicode package in XML/Ada support.

- **Touisteur**  
May 14th, 2014

Hi Pascal,

Are there already some examples of Adacore-sponsored libraries that use this great new tool ?

Any chance this will be used in future releases of AWS ? Would be great to be able to have the same build/install process on Win32/Linux.

I'm also curious to see how the installation of something like florist would be handled.

Keep up the Good work !  
Kezp

• **Pascal Obry**  
May 15th, 2014

Hi Kezp,  
Yes there is, AWS has already moved to using gprinstall. Same for wPOSIX, Templates\_Parser, Win32Ada.  
Others tools will be migrated in the future.

Regards,  
Pascal.