

Hunting for a C++ Packet Manager



Пакетные менеджеры уровня ОС

- FreeBSD Ports Collection;
- Advanced Packaging Tool;
- Debian package;
- Portage;
- Steam;
- Chocolatey;
- Homebrew;
- Apple Store;
- Google Play.

Пакетные менеджеры уровня ЯП

Java: Maven, Gradle, Maven Ant Tasks (**retired**).

Python: Pip.

Ruby: RubyGems.

JavaScript: Npm.

PHP: Composer.

RubyGems – для командной строки

```
~ gem list --local
```

```
*** LOCAL GEMS ***
```

```
bigdecimal (1.2.0)
```

```
json (1.7.7)
```

```
rake (0.9.6)
```

```
sqlite3 (1.3.7)
```

```
~ gem search xml
```

```
*** REMOTE GEMS ***
```

```
aastra_xml_api (1.1.4)
```

```
active_xml (0.0.4)
```

```
....
```

```
~ sudo gem install yaxml
```

```
Fetching: yaxml-0.1.gem (100%)
```

```
Successfully installed yaxml-0.1
```

```
...
```

```
1 gem installed
```

Bundler, GemFile – файл для проекта

```
source 'https://rubygems.org'
```

```
gem 'nokogiri'
```

```
gem 'rails', '3.0.0.beta3'
```

```
gem 'rack', '>=1.0'
```

```
gem 'thin', '~>1.1'
```

Maven как файл проекта

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany.app</groupId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <name>Maven Example</name>
  <dependencies>
    <dependency>
      <artifactId>junit</artifactId>
      <version>4.8.2</version>
    </dependency>
  </dependencies>
</project>
```

Как это работает для Maven

`/folder/with/maven/xml/file ~ mvn package -Dmaven.test.skip=true`

Или по другому:

`/folder/with/maven/xml/file ~ mvn eclipse:eclipse`

Между делом для windows, maven чаще всего выглядит как инородная программа на которую надо показать в PATH.

Мир C++ хорошо описывается
следующим постом на Reddit

Is there a C++ package manager? If not, how do you handle dependencies?

Мир C++ хорошо описывается следующим постом на Reddit

Is there a C++ package manager? If not, how do you handle dependencies?

- Welcome to the big leagues, where there are approximately 74 different build systems and a patchwork framework of 28 various package managers, most of which only support a certain platform/environment and are useless outside of it.

Мир C++ хорошо описывается следующим постом на Reddit

Is there a C++ package manager? If not, how do you handle dependencies?

- Welcome to the big leagues, where there are approximately 74 different build systems and a patchwork framework of 28 various package managers, most of which only support a certain platform/environment and are useless outside of it.

- HA HA HA HA

HA Ha ha ha **Please** help

Пакетных менеджеров в C++ достаточно много

Nuget

Conan

Cppget

Cget

Biicode

Qpm

Hunter

Gradle For C++

CPM

Pacm

Vcpkg

Чего хотелось бы от пакетного менеджера?

- **Кросс-платформенность;**
- Много пакетов, много хороших нужных пакетов (boost / json / xml / curl / openssl, zlib ...);
- Минимально возможный набор зависимостей;
- Простое использование, документация;
- Находящийся в разработке, а не в стадии архива или в стадии “ничего пока нет”;
- Быстродействие.

Проблемы с кроссплатформенностью и недостаточным набором пакетов

Nuget is the package manager for the Microsoft development platform including .NET.

Vcpkg helps you get C and C++ libraries on Windows.

Qml is a package manager for Qt.

Pacm has tiny amount of ready to use packets.

Cget seems to be only boost installer for a long time.

Cppget has 13 packages.

Кладбище

Biicode: "Sad to announce that we will be closing doors by 16th Feb, administrators have dictated servers shutdown." (c)

Кто же остался?

CPM - A C++ Package Manager based on CMake and Git.

Conan - can be installed in many Operating Systems. It is extensively used and tested in Windows, Linux (different distros), OSX, and also actively used in FreeBSD and Solaris SunOS, but it has been reported to work in other systems too.

Hunter - Cross-platform package manager for C++ (based on CMake ExternalProject). Supported platforms: Linux, Mac, Windows, iOS, Android, Raspberry Pi.

CPM – C++ Package Manager

- + Выглядит как менеджер с **наибольшим количеством пакетов**;
- + Есть веб-сайт где можно найти пакеты, оставить отзывы о пакетах;
- + У самого популярного пакета Caffe 12880 stars, 1598 subscribers;
- + Последний раз пакет был добавлен 23 марта 2017 года;
- + Методика установки выглядит как скрипт в 20 строк который надо добавить в CMake - это очень удобно для CMake проектов;
- Количество поддерживаемых “операционных систем”;
- Требование предустановленного GIT;
- Некоторые пакеты могут быть староваты.

CPM + Boost, что на входе

```
include(${CPM_DIR}/CPM.cmake)
```

```
CPM_AddModule("Boost"
```

```
    GIT_REPOSITORY "https://github.com/cogumbreiro/cpm-boost"
```

```
    GIT_TAG "1.59.0")
```

```
CPM_Finish()
```

```
...
```

```
target_link_libraries(${EXE_NAME} ${CPM_LIBRARIES})
```

CPM + Boost, что на выходе

```
Запуск #1      cmake ./ 91.58s user 54.61s system 11% cpu 20:46.22 total
Запуск #2+     cmake ./  3.25s user  5.14s system 35% cpu 23.625 total
```

Размер 1.3 GB.
Header-Only Boost

СРМ Конструкция

<projdir> /cpm-bin
 /cpm_packages/.git
 /modules/github_cogumbreiro_cpmboost_1590
←
/src/.git
 /boost/.git
 /libs/asio/.git
 /system/.git
 /tuple/.git
 /graph/.git



Conan is a portable package manager

- + Код добавляется часто, проект скорее жив чем мертв;
- + Кроссплатформенность / 2 ;
- + Возможность настройки проекта через conanfile.txt;
- + Возможность установки из консоли с помощью conan приложения;
- + Можно создать свой Conan сервер с необходимыми для установки файлами;
- Требуется Python (хотя в “тестовом” режиме достаточно brew / CMake);

Conan + Boost, что на входе

```
>> cat conanfile.txt
```

```
[requires]  
Boost/1.63.0@eliaskousk/stable
```

```
[generators]  
Cmake
```

```
>> cat CMakeLists.txt
```

```
include(${CMAKE_BINARY_DIR}/conanbuildinfo.cmake)  
conan_basic_setup()
```

```
add_executable(mymain main.cpp)  
target_link_libraries(mymain ${CONAN_LIBS})
```


Conan + Boost, что на выходе

Запуск #1 conan install . 5.17s user 3.99s system 24% cpu 37.810 total

Запуск #2+ conan install . 0.42s user 0.23s system 80% cpu 0.801 total

Размер 152 MB.

Запускать пришлось так:

```
DYLD_LIBRARY_PATH=$DYLD_LIBRARY_PATH:~/ .conan/data/Boost/1.63.0/eliaskousk/stabl  
e/package/799fada071d3f31a4af934c4bb02dc71da78a99b/lib/ ./bin/mymain
```


Conan + OpenSSL, что на входе

```
>> cat conanfile.txt
```

```
[requires]  
OpenSSL/1.0.2g@lasote/stable
```

```
[generators]  
Cmake
```

```
>> cat CMakeLists.txt
```

```
include(${CMAKE_BINARY_DIR}/conanbuildinfo.cmake)  
conan_basic_setup()
```

```
add_executable(mymain main.cpp)  
target_link_libraries(mymain ${CONAN_LIBS})
```

Conan + OpenSSL, что на выходе

Запуск #1 `conan install 70.27s user 17.62s system 85% cpu 1:42.98 total`

Запуск #2+ `conan install . 0.42s user 0.23s system 80% cpu 0.801 total`

Размер 264 MB. Скомпилированные на лету библиотеки.

(Boost лежит там же и занимает 152 MB).

На самом деле пришлось вызвать `conan install . --build OpenSSL`

Conan Конструкция

<projdir>/conanbuild.info - **conan CMake** пути к библиотекам и CMake макросы;
/conaninfo.txt - **настр. проекта** (опции библи., архитектура, компилятор).

~/.conan/conan.conf
/settings.yml
/data/OpenSSL/1.0.2g/lasote/stable
/data/Boost/1.63.0/eliaskousk/stable
←
/export/FindBoost.cmake
/package/799.someid.99b/FindBoost.cmake
/conaninfo.txt
/include/boost/*/*.hpp
/lib/libboost_*.dylib

Hunter – CMake-driven cross-platform package manager for C++

- + Платформы: Linux, Mac, Windows, iOS, Android, Raspberry Pi;
- + Мультиверсионность на одной машине для разных проектов;
- + По идеологии компилирует библиотеки на месте при возможности;
- + Есть список поддерживаемых пакетов;
- Чтобы использовать надо подсунуть CMake файл (можно добавить submodule в git);

Hunter + Boost, что на входе

```
>> cat CMakeLists.txt
```

```
cmake_minimum_required(VERSION 3.1.0 FATAL_ERROR)
include("HunterGate.cmake")
```

```
HunterGate(
    URL "https://github.com/ruslo/hunter/archive/v0.16.15.tar.gz"
    SHA1 "6974c2150fc0d3b09de3ad1efcbf15d360647ffa"
)
```

```
hunter_add_package(Boost COMPONENTS system filesystem)
find_package(Boost CONFIG REQUIRED system filesystem)
```

```
add_executable(mymain main.cpp)
target_link_libraries(mymain PUBLIC Boost::system Boost::filesystem)
```


Hunter + Boost, что на выходе

Запуск #1 cmake ./ 167.38s user 60.10s system 38% cpu 9:54.94 total

Запуск #2+ cmake ./ 1.11s user 0.62s system 79% cpu 2.186 total

Размер 202 MB (скомпилированные библиотеки Boost “на лету”) / вместе с исходным кодом Boost.

+ thread, chrono, date_time, regex, atomic, program_options, test, log.

Запуск #1 cmake ./ 1501.32s user 240.93s system 158% cpu 18:18.56 total

Запуск #2+ cmake ./ 1.76s user 1.43s system 81% cpu 3.921 total

Размер 417 MB (скомпилированные библиотеки Boost “на лету”) / вместе с исходным кодом Boost.

Hunter + OpenSSL, что на входе

```
>> cat CMakeLists.txt
```

```
cmake_minimum_required(VERSION 3.1.0 FATAL_ERROR)
include("HunterGate.cmake")
```

```
HunterGate(
    URL "https://github.com/ruslo/hunter/archive/v0.16.15.tar.gz"
    SHA1 "6974c2150fc0d3b09de3ad1efcbf15d360647ffa"
)
```

```
hunter_add_package(OpenSSL)
find_package(OpenSSL REQUIRED)
```

```
add_executable(mymain main.cpp)
target_link_libraries(mymain OpenSSL::SSL OpenSSL::Crypto)
```


Hunter + OpenSSL, что на выходе

Запуск #1 cmake ./ 74.84s user 21.52s system 83% cpu 1:55.82 total

Запуск #2+ cmake ./ 1.07s user 0.66s system 88% cpu 1.951 total

Размер 433 MB (скомпилированные библиотеки OpenSSL “на лету”) / вместе с исходным кодом OpenSSL.

(Boost лежит там же и занимает 417 MB).

Hunter Конструкция

<projdir>/_3rdParty/Hunter/install-root-dir - путь к каталогу библиотекам;
/config-id/config.cmake - версии библиотек по умолчанию;
/toolchain/

~/.hunter/_Base/Download/Boost/1.62.0/5fd9743/boost_1_62_0.tar.bz2
/Hunter/0.16.15/69754c21/v.0.16.15.tar.gz
/Unpacked/ ...

~/.hunter/_Base/Cache/meta/93b575e/Boost/ ...

~/.hunter/_Base/Cache/meta/93b575e/OpenSSL/ ...

←
~/.hunter/_Base/6974c21/78315bd/93b575e/

/Build/Boost/ ...

/Install/ - симуляция корневого <ROOT> каталога.

Спасибо!

Иван Сидоров

ivan.sidarau@gmail.com

Хорошие решения – результат опыта.

Опыт результат плохих решений. (с)