

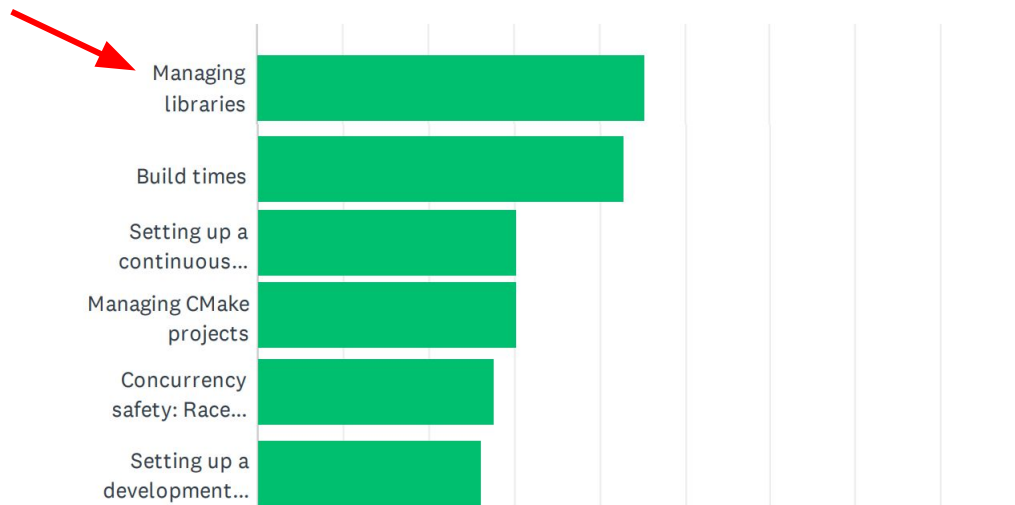
conan quirks in practice



C++ dependencies pain still going on

Q6 Which of these do you find frustrating about C++ development?

Answered: 1,261 Skipped: 4



isocpp.org/files/papers/CppDevSurvey-2024-summary.pdf

Conan advertisements

cppcon 
the c++ conference
SEPTEMBER 18-23 **2016**
Bellevue, Washington, USA

CONAN, C/C++ package manager

Presenter: Diego Rodriguez-Losada

What's New in Conan 2.0
Lessons Learned from the C++ Ecosystem

DIEGO RODRIGUEZ-LOSADA



**20
22** 
September 12th-16th



Common Package Specification (CPS) in practice

A Full Round Trip Implementation in Conan C++ Package Manager

DIEGO RODRIGUEZ-LOSADA GONZALEZ



**20
24** 
September 15 - 20



Conan advertisements

cppcon 
the c++ conference
SEPTEMBER 23-28 2018
Bellevue, Washington, USA

Git, CMake, Conan - How
to ship and reuse our C++ projects

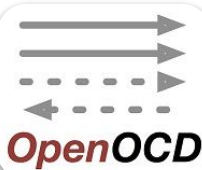
Presenter: Mateusz Pusz



Syntacore



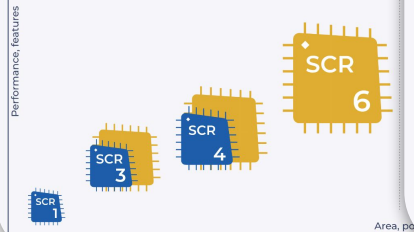
GDB
The GNU Project
Debugger



5

RISC-V Core IP

Microcontroller-class cores

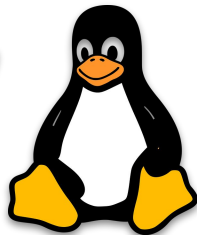
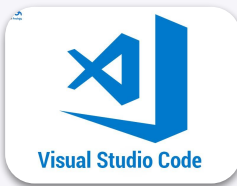


Software and Tools

Everything you need to develop
RISC-V applications. With
maintenance and support.



llvm-snippy



Syntacore™
Custom cores and tools

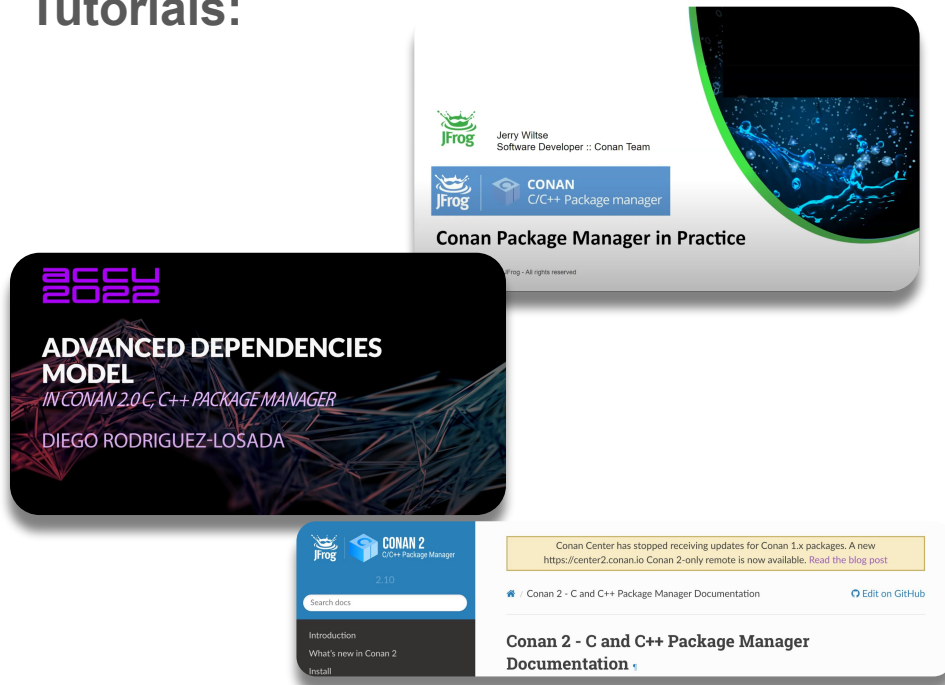


Disclaimer: this is not a tutorial

This talk is about:

- Non-obvious problems you meet integrating Conan
- How to deal with such problems

Tutorials:



Two sides of Conan integration

DevOps side

- Setup Python packages registry
- Setup Conan packages registry
- Setup 3rd party binaries backup registry
- Define global.conf configuration
- Define list of platforms and profiles

Conan side

- Write your recipes

DevOps integration: Conan internal registry

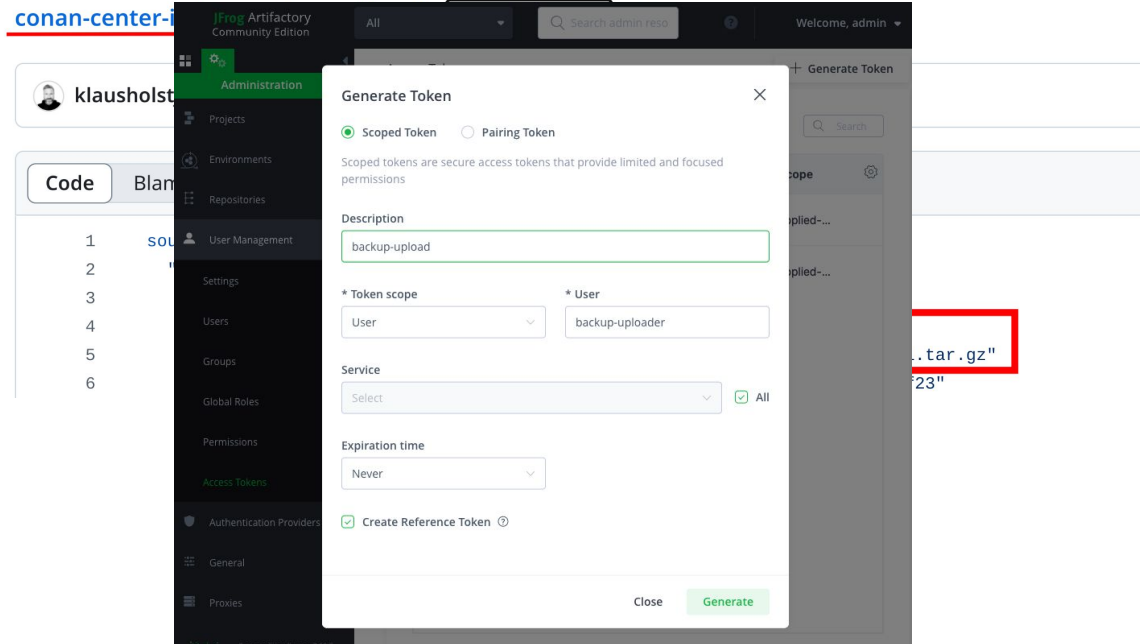


DevOps integration: PyPi internal mirroring registry



```
uv run conan ...  
my_python_tool ...
```

DevOps integration: 3rd party sources registry



docs.conan.io/2/devops/backup_sources/sources_backup.html

DevOps integration: global.conf

```
# ~/.conan2/global.conf
core.download:retry = 10
core.download:retry_wait = 4
core.download:parallel = 4 # conan >= 2.0.14
core.upload:retry = 10
core.upload:retry_wait = 4
core.sources:download_urls = ["origin", "https://artifactory.syntacore.com/artifactory/conan_os_backup"]
core.net.http:timeout = 420
core.net.http:max_retries = 8
tools.files.download:retry = 10
tools.files.download:retry_wait = 4
```

DevOps integration: profiles

```
# ~/.conan2/settings.yml
os:
  Windows:
    subsystem: [null, cygwin, msys, msys2, wsl]
  WindowsStore:
    version: ["8.1", "10.0"]
  WindowsCE:
    platform: ANY
    version: ["5.0", "6.0", "7.0", "8.0"]
  Linux:
  iOS:
    version: &ios_version
      ["7.0", "7.1", "8.0", "8.1", "8.2", "8.3", "9.0", "9.1", "9.2", "9.3", "10.0",
       "11.0", "11.1", "11.2", "11.3", "11.4", "12.0", "12.1", "12.2", "12.3", "12.4",
       "13.0", "13.1", "13.2", "13.3", "13.4", "13.5", "13.6", "13.7",
       "14.0", "14.1", "14.2", "14.3", "14.4", "14.5", "14.6", "14.7", "14.8",
       "15.0", "15.1", "15.2", "15.3", "15.4", "15.5", "15.6", "16.0", "16.1"]
```

DevOps integration: profiles

```
# ~/.conan2/settings.yml
```

```
os:
```

```
  Windows: ...
```

```
  Linux:
```

```
    distro:
```

```
      null:
```

```
      Ubuntu:
```

```
        version: [null, "14.04", "14.10", "15.04", "15.10", "16.04", "16.10",  
                  "17.04", "17.10", "18.04", "18.10", "19.04", "19.10",  
                  "20.04", "20.10", "21.04", "21.10", "22.04", "22.10",  
                  "23.04", "23.10", "24.04", "24.10", "25.04", "25.10"]
```

```
      CentOS:
```

```
        version: [null, ANY]
```

```
      Fedora:
```

```
        version: [null, "36", "37", "38", "39", "40", "41", "42", "43"]
```

```
      Rocky:
```

```
        version: [null, ANY]
```

```
  iOS:
```

```
    version: &ios_version
```

DevOps integration: profiles list

```
# ~/.conan2/profiles
```

```
sc_gcc9_centos7
```

```
sc_gcc13_centos7
```

```
sc_gcc9_ubuntu22
```

```
sc_gcc13_ubuntu22
```

```
sc_rv64
```

```
default -> sc_gcc13_ubuntu22
```

DevOps integration: beginner's guide

```
uv sync
source .venv/bin/activate
conan --version
conan config install conan-config.syntacore.com
```

Before Conan

```
wget artifactory.syntacore.com/tools/qemu/9.0.2.tar.gz
```

```
git submodule update --init --recursive
```

```
cmake -S . -B build --preset gcc_debug
```

Recipe example

llvm/conanfile.py

```
# type: ignore  
# pylint: disable=no-member,pointless-statement,invalid-name,not-callable
```

```
from conan import ConanFile
```

```
class Tool(ConanFile):
```

```
    name = "llvm"  
    version = "19.1.2"  
    user = "syntacore"  
    url = "https://gitlab.syntacore.com/llvm"
```

```
    settings = "arch", "os", "build_type", "compiler"  
    revision_mode = "scm"  
    package_type = "application"
```

```
    exports_sources = ["*", "!git/*", "!build/*"]
```

```
    def requirements(self):
```

```
        self.requires("sleef/[>=3.7, <4.0]")  
        self.requires("qemu/[>=9.0.0, <10.0.0]@syntacore",  
                      build=True, test=True)
```

```
    def build(self): ...
```

```
    def package(self): ...
```

```
    def package_info(self): ...
```

Install dependencies adventure

```
conan install .
```



Install dependencies adventure: lockfiles

===== Installing packages =====

ERROR: Missing binary: qemu/9.0.2@syntacore:3b1a8e58deb086c205bf5be8d78ceaa29153746a

qemu/9.0.2@syntacore: WARN: Can't find a 'qemu/9.0.2@syntacore' package binary
'3b1a8e58deb086c205bf5be8d78ceaa29153746a' for the configuration:

[settings]

arch=x86_64

os=Linux

os.distro=Ubuntu

os.distro.version=22.04

compiler=gcc

compiler.version=13

[requires]

libdwarf/0.11.1@syntacore#1c1ece463fef8cf6f9e3ea7b8976f97ec5d47216:da39a3ee5e6b4b0d3255bfef95601890afd807
ncurses/6.5.0@syntacore#4263c66873d69d619b51fc071ce515137d20866e:755a39c13ef664c0b348fe4871588387f80b02

ERROR: Missing prebuilt package for 'qemu/9.0.2@syntacore'

Check the available packages using 'conan list qemu/9.0.2@syntacore:* -r=remote'

or try to build locally from sources using the '--build=qemu/9.0.2@syntacore' argument

More Info at '<https://docs.conan.io/2/knowledge/faq.html#error-missing-prebuilt-package>'

Fixing conan.lock

```
# conan.lock
{
  "version": "0.5",
  "requires": [
    "sleef/3.7#f52e03ae3d251dec704634230cd806a2%1717689747.199",
  ],
  "build_requires": [
    "qemu/9.0.2#f52e03ae3d251dec704634230cd806a2%1717689747.199",
    "libdwarf/0.9.2@syntacore#13c96f538b52e1600c40b88994de240f%1682321681.826",
    "ncurses/6.2.0@syntacore#cf74779ae0103a9c29c0796fe4f4b7310cb3d9b%1732696022.699",
    "binutils/2.38#4d3f10e5192b471cc783990fe8994772%1723470330.599"
  ],
  "python_requires": [ ]
}
```

Side note: reproducibility



Side note: reproducibility



CLI / Configuring / package-lock.json

package-lock.json

A manifestation of the manifest

docs.npmjs.com/cli/v10/configuring-npm/package-lock-json



Committing your `poetry.lock` file to version control

python-poetry.org/docs/basic-usage

docs.astral.sh/uv/guides/projects/#uvlock

» Why have Cargo.lock in version control?

doc.rust-lang.org/cargo/faq.html#why-have-cargolock-in-version

Lockfiles

<https://docs.conan.io/2/tutorial/versioning/lockfiles.html>



CONAN

C/C++ Package Manager



Install dependencies adventure

```
conan install .
```

Install dependencies adventure: ABI control

===== Installing packages =====

ERROR: Missing binary: qemu/9.0.2@syntacore:3b1a8e58deb086c205bf5be8d78ceaa29153746a

qemu/9.0.2@syntacore: WARN: Can't find a 'qemu/9.0.2@syntacore' package binary
'3b1a8e58deb086c205bf5be8d78ceaa29153746a' for the configuration:

[settings]

arch=x86_64

os=Linux

os.distro=Ubuntu

os.distro.version=22.04

compiler=gcc

compiler.version=13

[requires]

libdwarf/0.9.2@syntacore#13c96f538b52e1600c40b88994de240f:da39a3ee5e6b4b0d3255bfef95601890afd807

ncurses/6.2.0@syntacore#cfdf74779ae0103a9c29c0796fe4f4b73:755a39c13ef664c0b348fe4871588387f80b02

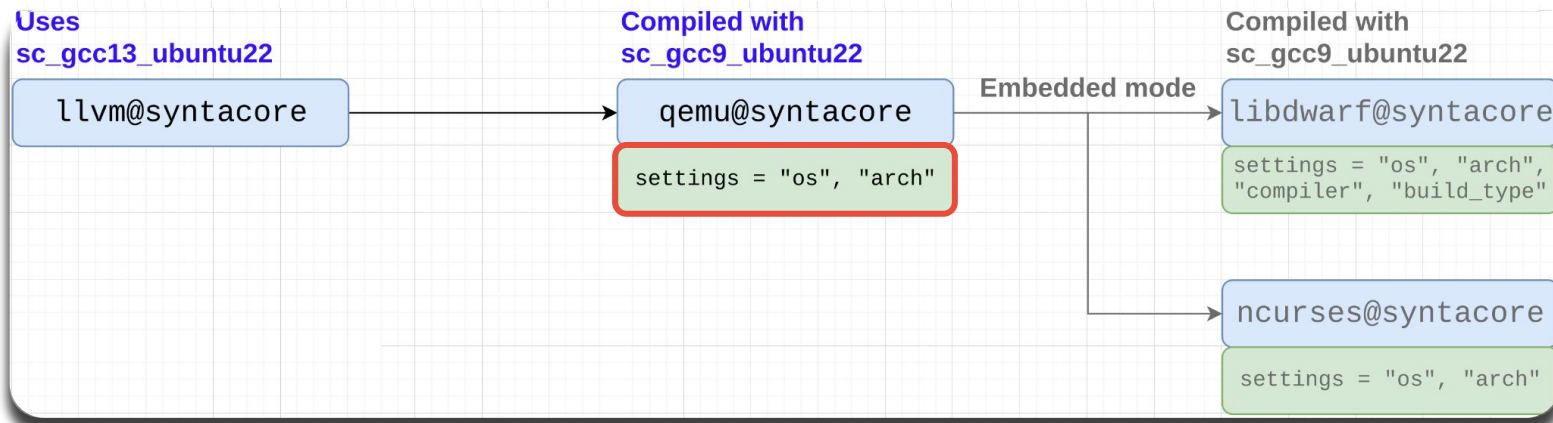
ERROR: Missing prebuilt package for 'qemu/9.0.2@syntacore'

Check the available packages using 'conan list qemu/9.0.2@syntacore:* -r=remote'

or try to build locally from sources using the '--build=qemu/9.0.2@syntacore' argument

More Info at '<https://docs.conan.io/2/knowledge/faq.html#error-missing-prebuilt-package>'

Install dependencies adventure: ABI control



package_id includes os, arch AND compiler!

Install dependencies adventure

```
conan install .
```

Install dependencies adventure: ABI control 2

===== Installing packages =====

ERROR: Missing binary: qemu/9.0.2@syntacore:dc6ee9874d89e81ac45f585c92bb0ffdfab88c2e

qemu/9.0.2@syntacore: WARN: Can't find a 'qemu/9.0.2@syntacore' package binary
'dc6ee9874d89e81ac45f585c92bb0ffdfab88c2e' for the configuration:

[settings]

arch=x86_64

os=Linux

os.distro=Ubuntu

os.distro.version=22.04

[requires]

libdwaf/0.9.2@syntacore#13c96f538b52e1600c40b88994de240f:da39a3ee5e6b4b0d3255bfef95601890afd807

ncurses/6.2.0@syntacore#cf74779ae0103a9c29c0796fe4f4b73:755a39c13ef664c0b348fe4871588387f80b02

ERROR: Missing prebuilt package for 'qemu/9.0.2@syntacore'

Check the available packages using 'conan list qemu/9.0.2@syntacore:* -r=remote'

or try to build locally from sources using the '--build=qemu/9.0.2@syntacore' argument

More Info at '<https://docs.conan.io/2/knowledge/faq.html#error-missing-prebuilt-package>'

Install dependencies adventure: ABI control 2

conaninfo.txt

package_id=\$(sha1sum conaninfo.txt)

```
[settings]
arch=x86_64
os=Linux
os.distro=Ubuntu
os.distro.version=22.04
```

[options]

...

[requires]

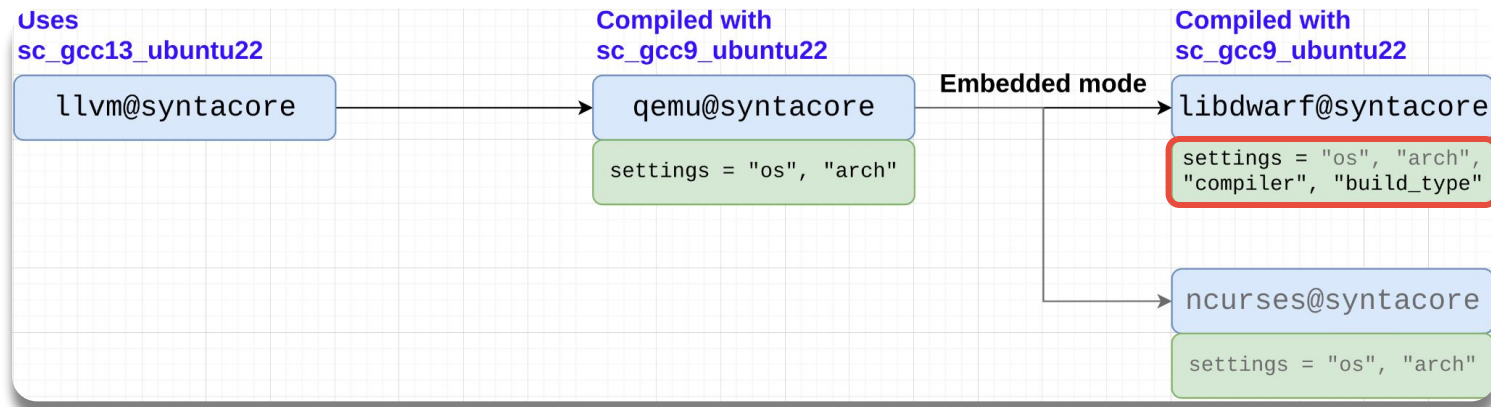
...

```
libdwarf/0.9.2@syntacore#313c96f538b52e1600c40b88994de240f:da39a3ee5e6b4b0d3255bfe95601890afd807b34
ncurses/6.2.0@syntacore#cf7d4779ae0103a9c29c0796fe4f4fb73:755a39c13ef664c0b348fe4871588387f80b02
```

recipe revision

package_id

Install dependencies adventure: ABI control 2



package_id includes libdwarf's package_id with compiler!

Install dependencies adventure: ABI control 2

```
# qemu/conanfile.py
def requirements(self):
    self.requires("libdwarf/[>=0.9.0, <2.0.0]", package_id_mode="revision_mode")
```

Install dependencies adventure

```
conan install .
```

Install dependencies adventure: unstable `package_id`

===== Installing packages =====

ERROR: Missing binary: qemu/9.0.2@syntacore:c72b988c74bd43b0d0cdc01d3308af9dcab512b8

qemu/9.0.2@syntacore: WARN: Can't find a 'qemu/9.0.2@syntacore' package binary
'c72b988c74bd43b0d0cdc01d3308af9dcab512b8' for the configuration:

[settings]

arch=x86_64

os=Linux

os.distro=Ubuntu

os.distro.version=22.04

[requires]

libdwarf/0.9.2@syntacore#13c96f538b52e1600c40b88994de240f

ncurses/6.2.0@syntacore#cf74779ae0103a9c29c0796fe4f4b73:755a39c13ef664c0b348fe4871588387f80b02

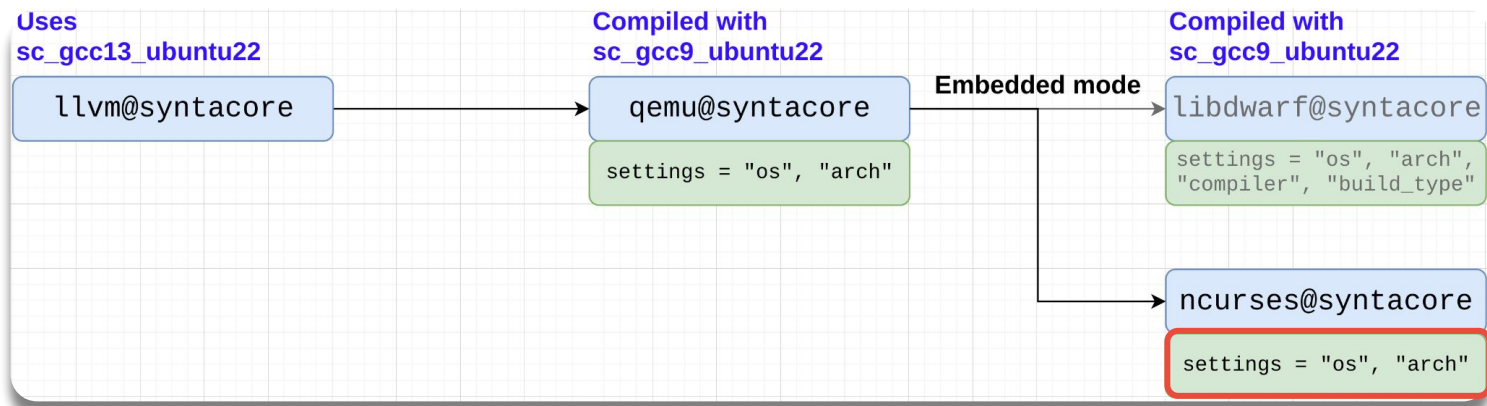
ERROR: Missing prebuilt package for 'qemu/9.0.2@syntacore'

Check the available packages using 'conan list qemu/9.0.2@syntacore:* -r=remote'

or try to build locally from sources using the '--build=qemu/9.0.2@syntacore' argument

More Info at '<https://docs.conan.io/2/knowledge/faq.html#error-missing-prebuilt-package>'

Install dependencies adventure: unstable `package_id`



Install dependencies adventure: unstable `package_id`

**CONAN**

C/C++ Package Manager

Stable

From Conan 2.0 and onwards, there is a commitment to stability, with the goal of not breaking user space while evolving the tool and the platform. This means:

- Moving forward to following minor versions 2.1, 2.2, ..., 2.X should never break existing recipes, packages or command line flows
- If something is breaking, it will be considered a regression and reverted.
- Bug fixes will not be considered breaking, recipes and packages relying on the incorrect behavior of such bugs will be considered already broken.
- Only documented features in <https://docs.conan.io> are considered part of the public interface of Conan. Private implementation details, and everything not included in the documentation is subject to change.
- The compatibility is always considered forward. New APIs, tools, methods, helpers can be added in following 2.X versions. Recipes and packages created with these features will be backwards incompatible with earlier Conan versions.
- Only the latest released patch (major.minor.patch) of every minor version is supported and stable.

Install dependencies adventure: unstable `package_id`

Commits

History for [conan](#) / [conans](#) / [client](#) / [graph](#) / `compute_pid.py` on `develop2`

All users

All time

Commits on Nov 4, 2024

refactor methods conans->conan (#17268)



memsharded authored on Nov 4 · ✓ 1 / 1

Verified

508ec9a



Commits on Oct 8, 2024

moved exceptions conans->conan (#17126)



memsharded authored on Oct 8 · ✓ 1 / 1

Verified

dcea8d9



Commits on Sep 24, 2024

Feature/build compatibles (#16871)



3 people authored on Sep 24 · ✗ 0 / 1

Verified

ea77696



Commits on Sep 23, 2024

refactor, avoid getting same core config for every package (#16962)



memsharded authored on Sep 23 · ✓ 1 / 1

Verified

95fa028



Install dependencies adventure

```
uv pip install conan==2.8.0  
conan install .
```

Install dependencies adventure: cache incompatibility

```
WARN: Upgrade cache from Conan version '2.0.9'  
WARN: Running 2.8.0 Cache DB migration to add LRU column  
ERROR: Could not complete the 2.8.0 DB migration. Please manually remove  
your .conan2 cache and reinstall packages  
ERROR: duplicate column name: lru  
  
conn.execute(f'INSERT INTO {self.table_name} '  
sqlite3.OperationalError: table recipes has 5 columns but 4 values were supplied  
  
ERROR: table recipes has 5 columns but 4 values were supplied
```



Install dependencies adventure

```
rm -rf ~/.conan2
```

```
conan config install conan-config.syntacore.com
```

```
conan install .
```

Install dependencies adventure

```
===== Finalizing install (deploy, generators) =====
```

```
conanfile.py (llvm/19.1.2@syntacore): Generating aggregated env files
```

```
conanfile.py (llvm/19.1.2@syntacore): Generated aggregated env files: ['conanbuild.sh', 'conanrun.sh']
```

Install finished successfully

Install dependencies adventure

```
source conanbuild.sh
```

```
qemu ...
```

```
qemu ...
```

```
~/conan2/qemu5ef5e82d8101b/p/bin/qemu: /lib64/libc.so.6:  
version `GLIBC_2.27' not found
```

Install dependencies adventure

```
conan install --profile build profile --host sc_centos7 .
```

```
conan
```

```
os7 .
```

```
conan >= 2.0.14
```



```
qemu --help
```

```
usage: qemu-riscv64 [options] program [arguments...]
```

```
Linux CPU emulator (compiled for riscv64 emulation) # happy qemu stuff
```

Dynamic profiles

docs.conan.io/2/reference/config_files/profiles.html#profile-rendering

```
# ~/.conan2/profiles/default

{% set compiler, version, compiler_exe = detect_api.detect_default_compiler() %}
{% set runtime, _ = detect_api.default_msvc_runtime(compiler) %}
[settings]
compiler={{compiler}}
compiler.version={{detect_api.default_compiler_version(compiler, version)}}
compiler.runtime={{runtime}}
compiler.cppstd={{detect_api.default_cppstd(compiler, version)}}
compiler.libcxx={{detect_api.detect_libcxx(compiler, version, compiler_exe)}}
```

-> renders to user's machine -> missing binary

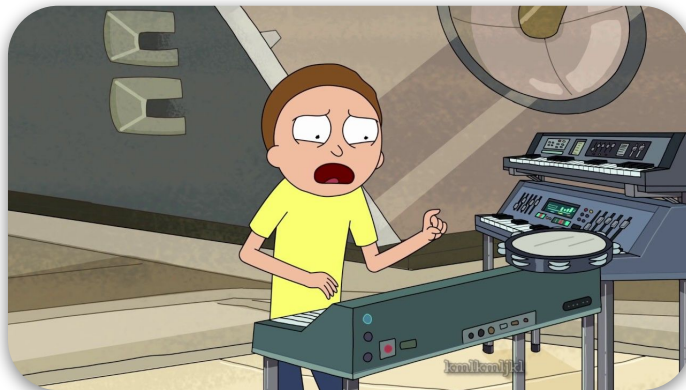
Dynamic profiles

```
# ~/.conan2/profiles/default
```

```
[settings]
arch={{ platform.machine() }}
build_type=Release
{% if platform.system() == "Linux" %}
    {% set raw_distro = os.popen("lsb_release -is").read().strip().lower() %}
    {% set raw_version = os.popen("lsb_release -rs").read().strip().lower() %}
    {% set distro = {"centos": "CentOS", "ubuntu": "Ubuntu"}.get(raw_distro, "Ubuntu") %}
    {% if distro == "CentOS" %}
        {% set version = {"7.9.2009": "7.9.2009"}.get(raw_version, "7.9.2009") %}
    {% elif distro == "Ubuntu" %}
        {% set version = {"18.04": "18.04", "20.04": "20.04", "22.04": "22.04"}.get(raw_version, "20.04") %}
    {% endif %}
    os=Linux
    os.distro={{ distro }}
    os.distro.version={{ version }}
{% elif platform.system() == "Windows" %}
    os=Windows
{% endif %}
```

Downloading packages adventure

```
wget artifactory.syntacore.com/tools/qemu/9.0.2.tar.gz
```



```
conan install --lockfile "" --output-folder output  
--deployer=full_deploy --requires qemu/9.0.2@syntacore
```

missing binary...

Downloading packages: vendoring dependencies

```
from conan import ConanFile

class Tool(ConanFile):
    name = "qemu"
    version = "9.0.2"
    user = "syntacore"
    url = "https://gitlab.syntacore.com/qemu"
    vendor = True # conan >= 2.4.0

    settings = "arch", "os", "build_type"
    revision_mode = "scm"
    package_type = "application"

    exports_sources = ["*", "!git/*", "!build/*"]

    def requirements(self):
        self.requires("libdwarf/[>=0.0, <1.0]@syntacore")
        self.requires("ncurses/[>=6.0.0, <7.0.0]@syntacore")
```

Downloading packages: vendoring dependencies

```
# conaninfo.txt
[settings]
arch=x86_64
os=Linux
os.distro=Ubuntu
os.distro.version=22.04
[options]
...
[requires]
...
libdwarf/0.9.2@syntacore#13c96f538b52e1c00c40b88994de240f
ncurses/6.2.0@syntacore#4263c66873d69d619b51fc071ce51513:755a39c13ef664c0b348fe4871588387f80b02
```

All customers get vendored package; you cannot allow some of them to recompile the package. You cannot even define vendoring based on some option.

Downloading packages: emulated vendoring

```
from conan import ConanFile
```

```
class Tool(ConanFile):
```

```
    name = "qemu"
```

```
    version = "9.0.2"
```

```
    user = "syntacore"
```

```
    url = "https://gitlab.syntacore.com/qemu"
```

```
    settings = "arch", "os", "build type"
```

```
    options = {"vendor": [True, False]}
```

```
    default_options = {"vendor": True}
```

```
    ...
```

```
    def requirements(self):
```

```
        vendor_args = {"package_id_mode": "unrelated_mode", "run": False,
                       "libs": False, "headers": True, "visible": False}
```

```
        kwargs = vendor_args if self.options.vendor else {}
```

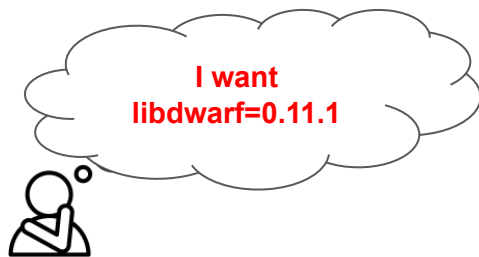
```
        self.requires("libdwarf/[>=0.0, <1.0]@syntacore")
```

```
        self.requires("ncurses/[>=6.0.0, <7.0.0]@syntacore")
```

Downloading packages adventure

```
conan install ... qemu/9.0.2@syntacore # Default deps
```

```
conan install ... qemu/9.0.2@syntacore -o vendor=False --build=qemu
```



1. Customers cannot precisely control libdwarf version
2. Hard to reuse binaries with custom libdwarf versions

Downloading packages adventure

```
conan install ... qemu/9.0.2@syntacore # Default deps
```

```
conan install ... qemu/9.0.2@syntacore -o libdwarf=0.11.1 --build=qemu
```

Second call on another machine

```
conan install ... qemu/9.0.2@syntacore -o libdwarf=0.11.1
```

Downloading packages: pin them all!

```
from conan import ConanFile
```

```
class Tool(ConanFile):
```

```
    name = "qemu"
```

```
    version = "9.0.2"
```

```
    user = "syntacore"
```

```
    settings = "arch", "os", "build_type"
```

```
    options = {"libdwarf": [None, "ANY"]}
```

```
    ...
```

```
    def requirements(self):
```

```
        kwargs = {"package_id_mode": "unrelated_mode", "run": False,
                  "libs": False, "headers": True, "visible": False}
```

```
        libdwarf = self.options.libdwarf or "0.9.2@syntacore#13c96f538b52e1600c40b88994de240f"
```

```
        self.requires(f"libdwarf/{libdwarf}", **kwargs)
```

```
    ...
```

Downloading packages adventure

```
conan install --lockfile "" --output-folder output  
--deployer=full_deploy --requires qemu/9.0.2@syntacore
```

docs.conan.io/2/reference/extensions/custom_commands.html

```
conan get qemu/9.0.2@syntacore
```



which one of `qemu/9.0.2@syntacore`?

Auto-versioning

```
conan list -r syntacore qemu/9.0.2@syntacore#*
```

```
syntacore
```

```
qemu
```

```
qemu/9.0.2@syntacore
```

```
revisions
```

```
b23663e2812cdca459b22bedb9f717bc5db1051a  
7c195649d708a906927a3251796f5b5cb92f727f  
6735dd11b3449a4a6481542ca9924864ca259bb6  
5cecc952a4e4ea092ed5aa544de474c7d9cdb4bc  
14b1370ef56b11e712bb66d224150c758a640ca4  
ab1d622297dabdb6a7476169993be5e9f116bd48  
1ed433da3a7180e9d3c309a2047babf972fe9cb8  
9322a76d0cd898b969ba62fc421961d1e9b2903c  
1849071ef363edbdab3c09bf28f0254f14268663  
dbf16bca221cc9cf79474ea81428e8ea7f544330  
76aeb2d90507e54848bdaa06203c735823986fb7  
3b120a86767794d935b11db09c3d78a4923ea772  
cb34efeda811c27ce379e948de7ade16fc48e33b  
...
```

Auto-versioning

```
# conanfile.py
def set_version(self):
    base = "9.0.2"
    anchor = "338bd65ac045f79d64db624bd7cb4a1db4893027"
    build = Git(self).run(f"rev-list {anchor}..HEAD --count --first-parent")
    branch = Git(self).run(f"rev-parse --abbrev-ref HEAD")
    self.version = f"{base}-{build}+{branch}"
```

`conan get qemu/9.0.2-42+main@syntacore`

`conan get qemu@syntacore`

Source code fetch



Source code fetch

```
from conan import ConanFile
```

```
class Tool(ConanFile):  
    name = "llvm"  
    user = "syntacore"  
    url = "https://gitlab.syntacore.com/llvm"  
  
    settings = "arch", "os", "build_type", "compiler"  
    revision_mode = "scm"  
    package_type = "application"
```

```
# Sources are located in the same place as this recipe, copy them to the recipe  
exports_sources = ["*", "!git/*", "!build/*"]
```

...

docs.conan.io/2/tutorial/creating_packages/create_your_first_package.html



Direct integration into source code

```
from conan import ConanFile
from conan.tools.scm import Git
```

```
class Tool(ConanFile):
```

```
...
```

```
def export(self):
```

```
    git = Git(self, self.recipe_folder)
```

```
    # save the url and commit in conandata.yml
```

```
    git.coordinates_to_conandata()
```

Saves only 1 remote:
CI downloads sources twice!

```
def source(self):
```

```
    # we recover the saved url and commit from conandata.yml and use them to get sources
```

```
    git = Git(self)
```

```
    git.checkout_from_conandata_coordinates()
```

docs.conan.io/2/examples/tools/scm/git/capture_scm/git_capture_scm.html

Direct integration into source code

```
from pathlib import Path
from conan import ConanFile
from conan.tools.scm import Git
from conan.tools.files import update_conandata

class Tool(ConanFile):
    ...
    def export(self):
        git = Git(self, self.recipe_folder)
        commit, local = git.get_commit(), git.get_repo_root()
        update_conandata(self, {"commit": commit, "local": local, "remote": self.url})

    def source(self):
        git = Git(self)
        coords = self.conan_data
        if Path(coords["local"]).exists():
            git.clone(url=coords["local"], target=".")
        else:
            git.clone(url=coords["remote"], target=".")
        git.checkout(coords["commit"])
```

Dark magic with python_requires_extend

```
from conan import ConanFile
```

```
class Tool(ConanFile):
```

```
    name = "qemu"
```

```
    version = "9.0.2"
```

```
    user = "syntacore"
```

```
    settings = "arch", "os", "build_type"
```

```
    python_requires = "sc_helper@syntacore"
```

```
    python_requires_extend = "sc_helper.SyntacoreConanFile"
```

```
    custom_versions = {"libdwarf", "expect"}
```

```
    ...
```

```
    def requirements(self):
```

```
        self.requires("libdwarf")
```

```
        self.requires("ncurses")
```

Auto auto-versioning

And other fixes...

Smart source code fetch

Fixed vendoring

<https://github.com/syntacore>



Summary

- | | |
|---|--|
| 1. Third-party resources blockage or removal | Setup internal 3rd party resources mirroring registry |
| 2. Missing Linux distros | Define Linux distros in <code>settings.yaml</code> OR build distro-independent |
| 3. Binaries consumption | Correctly define <code>settings</code> & use vendoring |
| 4. Binaries consumption: unstable <code>package_id</code> | Strictly control conan version and <code>package_id</code> compute |
| 5. Cache incompatibility | Wrap conan with custom runner |
| 6. Explicit profiles in CLI | Define smart default profile |
| 7. Binaries consumption: cumbersome commands | Define custom conan command for downloading |
| 8. Unreadable revision list for one version | Auto-version your packages |
| 9. Registry excessively filled with source code | Use <code>coordinates_to_metadata</code> tech or similar |

What we missed

1. Advanced `python_requires_extend` usage
2. Cache concurrency problem
3. Cyclic dependencies
4. Outdated build system files after ``git pull``
5. Multi-configuration updates
6. Experimental packages promotion
7. Consuming different configurations of the same package at once

Questions?

