Shipping ready-to-run Python apps without the need to install Python

Introducing eGenix PyRun

EuroPython 2024 – 11.07.2024

Prague, Czech Republic

Marc-André Lemburg :: eGenix.com GmbH

Speaker Introduction

Marc-André Lemburg

- Studied Mathematics, Computer Science and Physics
- CEO eGenix.com GmbH
- Senior Solution Architect, Consulting CTO and Coach
- EuroPython Society Fellow and former Chair
- Python Software Foundation Fellow and former director
- Python Core Developer (Unicode, DB-API, platform module, ...)
- Co-founder Python Meeting Düsseldorf
- Based in Düsseldorf, Germany

- More details and contact: https://malemburg.com/

Use case: Shipping Python products to end users

- Back in 2008: Shipping a server product using Python

    - Product: *eGenix mxOBDC Connect Server*
    - Needed a solution for shipping on Unix and Windows

    - Windows: py2exe solved the problem

    - Unix: Did not want to rely on OS version of Python
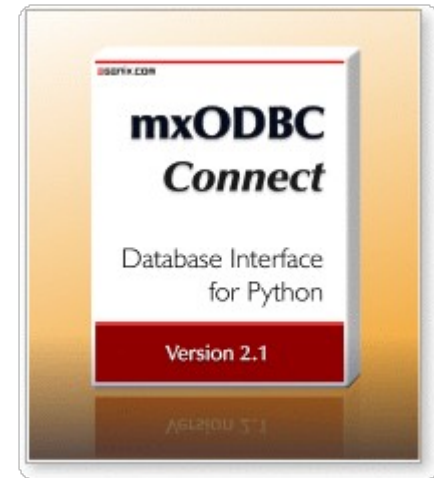
eGenix PyRun: Business Requirements in 2008

- Goals:
  - Have an easy to distribute Python runtime
  - Independent of OS Python installation
  - Ship products as byte compiled archives

- Target platforms:
  - Linux
  - macOS, FreeBSD
  - nice to have:
    other Unix variants

PyRun

Remembered mxCGIPython...

- Original project: mxCGIPython

  - First release 1998
  - Webhosters did not support Python

  Older Projects

23.01.00	platform.py	- standard module for querying platform information
28.04.99	mxCGIPython	- single file Python interpreter, good for doing CGI even without Python support from your ISP
25.02.99	py2html.py	- Python script colorizer (producing not only HTML pages)
26.09.98	doc.py	- Python doc-string to HTML converter

  - Idea: Install Python via FTP upload to cgi-bin directory
  - To keep things simple: Put the Python runtime into a single file

  - Lot's of interest, many contributions at the time
  - Supported Python 1.5.2, 2.1, 2.2, 2.3
  - Project slowed down in the early 2000s, stopped 2003

Getting everything into a single executable

- Main recipe

  - Statically build most stdlib Python C extensions

    - works using Modules/Setup[.local]
    - add any custom C extensions, statically linked

  - Put byte code into C arrays for most of the stdlib Python modules

    - add any custom Python packages to the set

  - Statically link everything into the executable

Tool behind encoding Python modules into C

- Main tool: Tools/freeze
  - Has been part of Python since the very early days (1994)
    … long before ZIP file imports were added
  - Written by Guido van Rossum
  - Later extended by Mark Hammond to also work on Windows

- What it does...
  - Takes a set of Python modules
  - Compiles them to byte code
  - Puts byte code into C arrays
  - Statically links the code to CPython

Evolution takes time

- *mxCGIPython* worked great to ship a single application, but was not very generic

- Challenge: Emulate the Python command line
  - Rewrote the C command line interface in Python
  - Tricky, due to many (sometimes undocumented) features
  - pip and setuptools rely on some of those features
  - A bit slower than regular Python command line interface

- Added support for interactive and ZIP file use

- Added support for pip and compiling C extensions

- Open sourced all this in 2012 as *eGenix PyRun*

Where we are today with eGenix PyRun

- Apache licensed, free drop-in Python runtime

- Available for Python 3.8 – 3.11

- Executable sizes around 4MB – 6MB

- Works on Linux and WSL
  - On macOS and FreeBSD probably as well (not tested)
  - Doesn't support native Windows (but on the roadmap)

- Includes most of the Python Stdlib
  - Supports package tooling, e.g. pip, setuptools, build
  - … and Python C extensions, such as Numpy, Pandas, Polars, etc.

PyRun

eGenix PyRun for Python 3.11

- PyRun is a Python Runtime in a single file

```
total 67M
lrwxrwxrwx 1 lemburg lemburg    9 Jun 26 12:34 pyrun -> pyrun3.11
-rwxr-xr-x 1 lemburg lemburg 5.5M Jun 26 12:33 pyrun3.11
-rwxr-xr-x 1 lemburg lemburg  43M Jun 26 12:33 pyrun3.11-debug
-rwxr-xr-x 1 lemburg lemburg  19M Jun 26 12:33 pyrun3.11-standard
lrwxrwxrwx 1 lemburg lemburg    9 Jun 26 12:34 pyrun3.11-upx -> pyrun3.11
lrwxrwxrwx 1 lemburg lemburg    9 Jun 26 12:34 python3 -> pyrun3.11
lrwxrwxrwx 1 lemburg lemburg    9 Jun 26 12:34 python3.11 -> pyrun3.11


newton pyrun-installation/bin> ./pyrun3.11
eGenix PyRun 3.11.9 (release 2.5.0, main, Jun 26 2024, 12:33:00) [GCC 7.5.0]
Thank you for using eGenix PyRun. Type "help" or "license" for details.

>>> █
```

What's in the box ?

- Almost all of the Python Stdlib

- Not included in the default build:
  - Test modules / packages
  - ensurepip, tkinter, turtle, parser, tabnanny
  - distutils, idlelib, smtpd, asyncore, asynchat

- Built as shared modules in the default build:
  - _decimal, ctypes, readline, curses, crypt, audioop, osaudiodev, spwd, _uuid

- All of this can be configured

eGenix PyRun: Updated Business Requirements in 2024

- Goals:
  - Ship whole apps as a single binary (or only few files)
  - Have an instant production ready Python environment
  - Independent of OS Python installation
  - Make all this as easy as possible

- Target platforms:
  - Linux, WSL
  - macOS, FreeBSD
  - eventually: native Windows, WASM, perhaps even mobiles

Other use cases of eGenix PyRun

- **Great for Docker containers** due to small footprint

  - We want to investigate creating an Alpine based
    container distribution for PyRun

- Truly OS independent venv replacement

  - Easily move your venv around
    by simply copying it to a new location

- Training and education

- Demos

  … probably lots more

Where we want to go from here

- Start a community around PyRun

- Make the code more accessible

- Make PyRun easier to use and (even) more versatile

- Relaunched project on July 1 with version 2.5.0
  - Extracted code from internal monorepo
  - Put everything on Github
  - Added Python 3.11 support
  - Python 3.12 will follow soon

PyRun

Advanced topics: Quickly building a single file Python app

- With the integrated ZIP file support, this is easy:

    - Create a Python app ZIP file with a __main__.py module
      to define the app's entry point

    - Run

      ```
      cat pyrun3.11 myapp.zip > hello
      chmod 755 ./hello
      ```

    - Done.

- PS: No compilation needed !

Advanced topics: Customizing PyRun

- Required skill set …

  You don't need to be a core developer to get this working :-)

- Know how Tools/freeze works
  - Adding Python packages is very easy
  - Excluding modules is a little harder:
    - edit Makefile EXCLUDES variable

- Know how Modules/Setup[.local] works
  - Only needed for adding new stdlib modules or custom C modules
  - Very simple Makefile style DSL

Advanced topics: Quickly building a single file Python app

- With the integrated ZIP file support, this is easy: ☺

  - Create a Python app ZIP file with a __main__.py module
    to define the app's entry point

  - Run

    ```
    cat pyrun3.11 myapp.zip > hello
    chmod 755 ./hello
    ```

  - Done.

- PS: No compilation needed !

FORGET THIS

Advanced topics: Customizing PyRun

- Required skill set …

  You don't need to be a core developer
  to get this working :-)

- Know how Tools/freeze works

  - Adding Python packages is very easy
  - Excluding modules is a little harder:
    - edit Makefile for MODULES variable

- Know how Modules/Setup[.local] works

  - Only needed for adding new stdlib modules or custom C modules
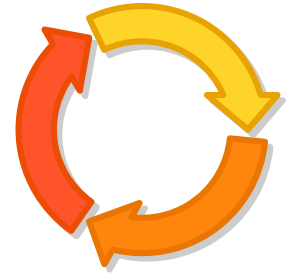 - Very simple Makefile style DSL

Interlude: eGenix PyRun CLI

- **We're currently working on making all this easy**

- **The CLI will (eventually) support:**

  - Installing / Updating / Building PyRun

  - Creating complete PyRun Apps (pyrun + application code)
    - As ZIP files or frozen as well
      … including C extensions !

  - Creating lean PyRun virtualenvs
    - Complete runtimes built from your site-packages or pyproject.yaml

Advanced topics: Porting eGenix PyRun to Python 3.x+1

- Usually takes a few days to weeks depending on what changed in CPython

- Ports tend to take longer with more recent releases
  - Python 3.8:   1-2 days   pyrun size: 4.8MB
  - Python 3.9:   2-3 days   pyrun size: 4.9MB
  - Python 3.10:   3 days   pyrun size: 4.8MB
  - Python 3.11:   7 days   pyrun size: 5.5MB

- New versions typically end up requiring more disk space

 … but on the plus side, also tend to be faster

Advanced topics: Porting eGenix PyRun to Python 3.x+1

- Usually takes a few days to weeks depending on what changed in CPython

- Ports tend to take longer with more recent releases
  - Python 3.8:   1-2 days   file size: 4.8MB
  - Python 3.9:   2-3 days   file size: 4.9MB
  - Python 3.10:   3 days   file size: 4.8MB
  - Python 3.11:   days   file size: 5.5MB

- New version typically end up requiring more disk space
 - … but on the plus side, also tend to be faster

THIS IS ON US ☺
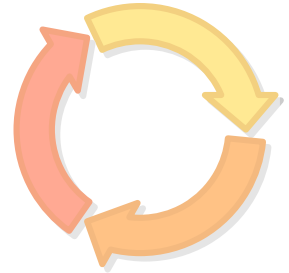
Contributions are welcome !

- We could use some help in the following areas:

  - Building and testing on various platforms

  - Domain expertise on macOS, FreeBSD and Windows

  - Exploring cross-compilation builds using Zig CC

  - Exploring new use cases

  - Spreading the word and growing a community

Main takeaway: Never stop **learning** and **trying out new things**
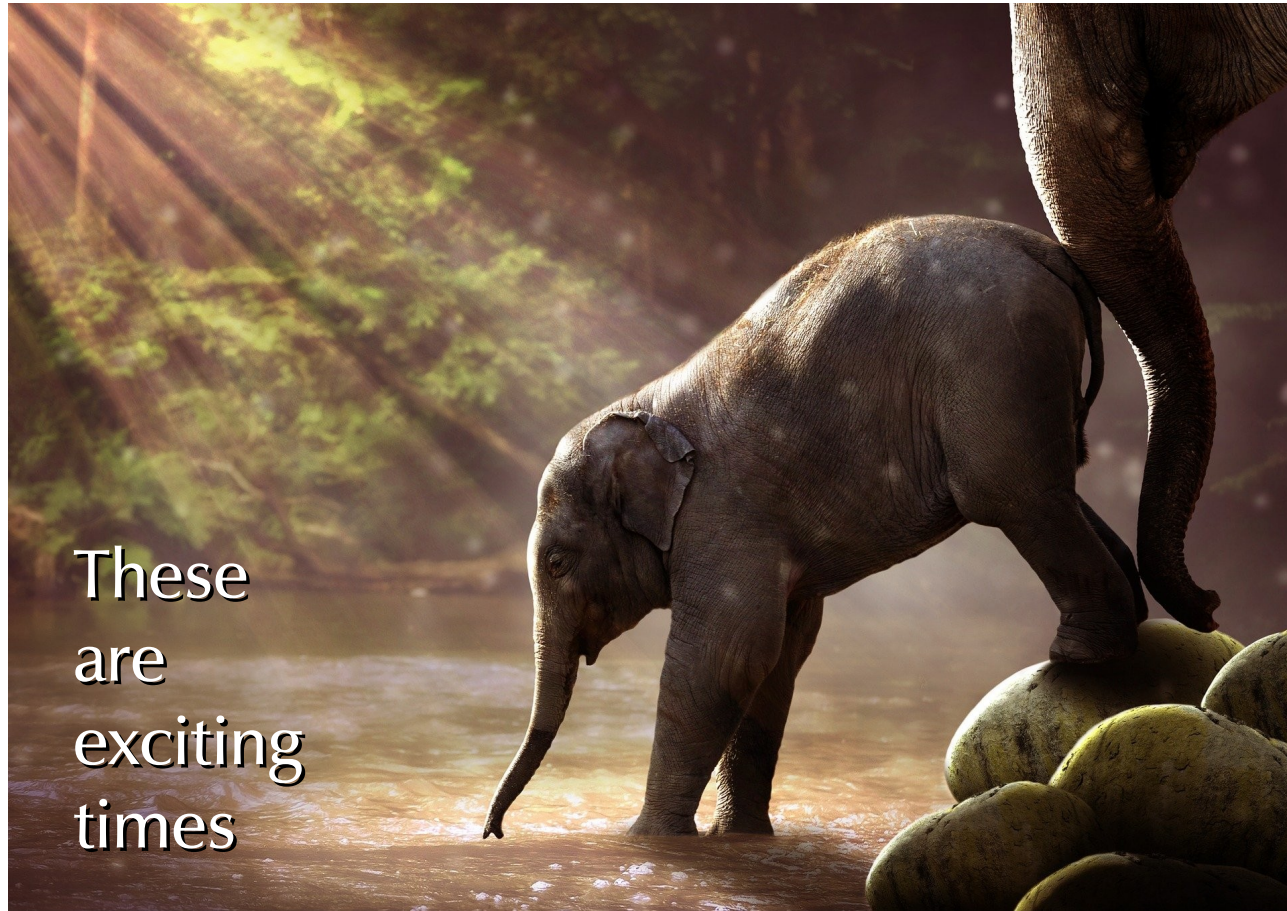
These are exciting times

Resources

- eGenix PyRun:

  - Product page:
    https://pyrun.org/

  - Project on Github:
    https://github.com/eGenix/egenix-pyrun

- Talk slides and resources:

  - Github:
    https://github.com/eGenix/egenix-pyrun-talk-resources

Thank you for your attention !

Time for discussion

Contact

eGenix.com Software, Skills and Services GmbH

Marc-André Lemburg

Pastor-Löh-Str. 48

D-40764 Langenfeld

Germany

eMail: mal@egenix.com

Phone: +49 211 9304112

Fax: +49 211 3005250

Web: https://www.egenix.com/

LinkedIn:

References

- Several photos taken from Pixabay and Unsplash

- Some screenshots taken from the mentioned websites

- All other graphics and photos are (c) eGenix.com or used with permission

- Details are available on request

- Logos are trademarks of their respective trademark holders