It contains a few Bash Scripts, which can help one install open source developer tools on OS X.

493 commits | 1 branch | 0 packages | 0 releases | 1 contributor | GPL-3.0

Branch: master | New pull request | Find file | Clone or download

drikosev Upgrade libiconv to 1.6 — Latest commit 2f298ab 1 minute ago

LICENSE	Initial commit	3 years ago
README	Upgrade libiconv to 1.6	1 minute ago
README-2019-12-08.pdf	Add files via upload	5 months ago
README-2019-12-09.pdf	Add files via upload	5 months ago
gfc-2018-10-11.tar.bz2	gfc: accept quoted literals	2 years ago
gfc-2020-04-28.tar.bz2	columns & full path names	13 days ago
pc-rules-2019-12-06.tar.bz2	Mojave aware	5 months ago
pc-rules-2019-12-08.tar.bz2	Few Enhancements	5 months ago
pc-rules-2019-12-10.tar.bz2	Spelling Errors & links	5 months ago
pc-rules-2019-12-11.tar.bz2	gcc-4.8.5 - backport of PR/83149	5 months ago
pc-rules-2019-12-12.tar.bz2	gcc-4.8.5 - backport of PR/33430 (partial)	5 months ago
pc-rules-2019-12-14.tar.bz2	Regression Fix	5 months ago
pc-rules-2019-12-21.tar.bz2	gcc-4.8.5 - backport of PR/80392	5 months ago
pc-rules-2019-12-26.tar.bz2	gcc-4.8.5 - backport of PR/83864	5 months ago
pc-rules-2020-01-14.tar.bz2	OCaml Configuration	4 months ago
pc-rules-2020-01-17.tar.bz2	OCaml Configuration	4 months ago
pc-rules-2020-01-22.tar.bz2	OCaml bindings in LLVM (3.4)	4 months ago
pc-rules-2020-02-05.tar.bz2	gcc-4.8.5 - backport of PR/54613 (partial)	3 months ago
pc-rules-2020-02-06.tar.bz2	gcc-4.8.5 - backport of PR/54613 (partial)	3 months ago
pc-rules-2020-02-07.tar.bz2	gcc-4.8.5 - backport of PR/54613 (linux)	3 months ago
pc-rules-2020-02-11.tar.bz2	random failure	3 months ago
pc-rules-2020-02-29.tar.bz2	gcc-4.8.5 - backport of PR/92785	2 months ago
pc-rules-2020-04-04.tar.bz2	GNU GCC 8.4.0	last month
pc-rules-2020-04-11.tar.bz2	gcc-4.8.5 - backport of PR/82886	29 days ago
pc-rules-2020-04-29.tar.bz2	gcc-4.8.5 - backport of PR/44672	10 days ago
pc-rules-2020-05-05.tar.bz2	gcc-4.8.5 - backport of PR/92976	5 days ago
pc-rules-2020-05-10.tar.bz2	gcc-4.8.5 - backport of PR/80118	5 minutes ago
pc2gfortran.spec	Build RPM - CentOS-7.6	10 days ago
runtests-2018-04-15-x86_64-apple-darwi...	+2 failures on 10.13.4	2 years ago
runtests-2019-02-26-x86_64-darwin17-gc...	gcc 4.8.5~36 (rel. 36 beta porting) +PRs ...	14 months ago
runtests-2019-03-05-x86_64-darwin17-g...	gcc 4.8.5~36 (rel. 36 beta porting) +PRs ...	14 months ago
runtests-2019-03-10-x86_64-darwin15-gc...	gcc 4.8.5~36 (rel. 36 beta porting) +PRs ...	14 months ago
runtests-2019-11-07-x86_64-darwin15-gc...	gcc 4.8.5~36 (rel. 36 beta porting) +PRs ...	6 months ago
runtests-2019-12-01-3.10.0-957.el7.x86_6...	gcc-4.8.5~39 (rel. 39 beta porting CentOS 7.6)	5 months ago
runtests-2019-12-03-x86_64-darwin18-gc...	gcc 4.8.5~39 (rel. 39 beta porting)	5 months ago
runtests-2020-04-05-3.10.0-957.el7.x86_...	GCC 8.4.0 on CentOS-7.6	last month
selective-gcc-tests.txt	Status of gfortran-4.8.5	3 minutes ago

README

```
Port Center
-----------

The port center (pc) is an OS directory that contains a few Bash Scripts and Make Files,
which can help one install open source developer tools on a Mac; formerly OS X, now macOS.

The Port Center automates also the installation of RPM, a package management system which
however requires a long list of dependencies to be installed first. The PC script runs on
OS X (10.9-10.11) and macOS (10.12-10.14). Yet, support for Mojave is work in progress.

Also, one can run this script in a RHEL 7.4-7.6 compatible system to install two Linux
Drivers, one for the Apple USB Super Drive (ausden) and another for the Broadcom IEEE
802.11a/b/g/n WiFi card (bcm43xx) installed on older Mac Computers (ie a Mac Mini 2011).
The WiFi Driver is kernel specific and has to be reinstalled after a kernel upgrade.


Installation Script
-------------------

To install a package, one has to run the "port" command to create an archive that will be
processed by the system installer. This "port" command cannot uninstall packages and
shouldn't be on path.

The installation process is logically divided in three phases. At first, one can run the
command "./port details <package>" to examine in advance the installation details of a
package, without download it; this command will print the relevant url's, configuration
options, and installation paths for that package.

After, one can archive a package with a single command or on a step-by-step basis.
Specifically, one can download, extract, patch, configure, make, place, pack, and archive
a package. These steps are also options of the port command. Running the "port" command
with the option "place" will install files to a temporary directory located at
"/tmp/<package>.dst"; which is the destination root that will be packed and archived.

Finally, one can invoke the system installer to process the archive; possibly, by double
clicking it. If you build and install a package on the same machine you can simply run:
"./port install <package>".

Once the RPM has been installed, one can use in the port script the command line arguments
"rpmbuild" and "rpminstall" instead of "archive" and "install" respectively for any open
source package. To see more type: "~/pc/port --help"


Download & Setup
----------------

Once the tarball has been downloaded (ie at ~/Downloads/pc-rules-2020-05-10.tar.bz2), the
following four commands will setup the port center:

install -d ~/pc
cd ~/pc
tar -xjSf ~/Downloads/pc-rules-2020-05-10.tar.bz2
ln -sf rules/port port


*SHA1 pc-rules-2019-12-06.tar.bz2: 29f65de5e10cf0dbb8152a8f9812482d3d2b5199
*SHA1 pc-rules-2019-12-08.tar.bz2: 53a2e20482a3b14809bc65d67c07434a41779711
*SHA1 pc-rules-2019-12-10.tar.bz2: 08d3b0b8e564afd40f636fc93af2843bc2feb613
*SHA1 pc-rules-2019-12-11.tar.bz2: aa73914afb92c49a485eed29be8d0a8ba22b9885
*SHA1 pc-rules-2019-12-12.tar.bz2: 8e6a0a0808617d4292414f42bb52f95c9da38c8e
*SHA1 pc-rules-2019-12-14.tar.bz2: 883acc25e81b603800bbf14fdd33fb59708ba2e1
*SHA1 pc-rules-2019-12-21.tar.bz2: d3433b7b5e32157648198bf756a06de381caf303
*SHA1 pc-rules-2019-12-26.tar.bz2: 195ab575298498de079382a9b0df154628f4b6f5
*SHA1 pc-rules-2020-01-14.tar.bz2: 31a9421c2ed56c18331151bd5ff6c11d26964d55
*SHA1 pc-rules-2020-01-17.tar.bz2: 6d0e63ff151228c668b3b3521499534a3824b23c
*SHA1 pc-rules-2020-01-22.tar.bz2: 7ba756ace30c3c71677378a907ec9c83bf5afbaa
*SHA1 pc-rules-2020-02-05.tar.bz2: c28cc759d0ca405dadbe1ef3de7e2f6e3406d2f9
*SHA1 pc-rules-2020-02-06.tar.bz2: cd3314c43dc86dee11ce9c955ea715364421449f
*SHA1 pc-rules-2020-02-07.tar.bz2: 2b92460efe73b78462766da874bbfe7976174ac2
*SHA1 pc-rules-2020-02-11.tar.bz2: 0594fd981611afd79da2813f1f0d6f4f19ee4163
*SHA1 pc-rules-2020-02-29.tar.bz2: e38da6ca6bdb79f22df62ecdd9f9fc36658b904d
*SHA1 pc-rules-2020-04-04.tar.bz2: 7bf09f61eb34834e7d8d9b826961eb9b03dea798
*SHA1 pc-rules-2020-04-11.tar.bz2: 274868d35b8077a183b7eb4c9e9910b77acf2488
*SHA1 pc-rules-2020-04-29.tar.bz2: 6e74dc33bddf5f8ec061a247ef2fc58b6a760dc1
*SHA1 pc-rules-2020-05-05.tar.bz2: c38b8ca636432d495eb7fdc2b3c813c2d6172081
*SHA1 pc-rules-2020-05-10.tar.bz2: 388afc89fd5ff3547a67ae2e96aecaf43ad25a1a

To see ie the build instructions for gcc-4.8.5, type:

./port details gcc4

Likely, this is the first package to be installed on MacOS (10.14) and a night build is
recommended. The file "selective-gcc-tests.txt" explains some build options about this
package. On macOS Mojave (10.14) the SDK Headers are required, see the known issues below.



RPM (4.11.3)
------------

[2017-11-13]
The RPM Package Manager (RPM) is a command line driven package management system used in
various Linux distributions, such as Red Hat Enterprise Linux, and SUSE Linux Enterprise.
According to wikipedia, it has also been ported to other systems such as IBM's AIX .

This package builds the fork distributed by rpm.org and now uses gnupg-2.0. This porting
is still work in progress and hasn't passed yet adequate quality tests. In example, we
haven't tested at all the RPM Plug-In functionality. If you face any problems with the
Berkeley DataBase, configure it with the minimum options possible.

To install this package and initialise the RPM DataBase, run:

./port install rpm

If the above command fail, follow a step by step approach as you may have to do with any
other package that fails, ie "autogen" on Mojave. Post a clean/cleaner command, configure
again, and then install:

./port clean rpm
./port -v configure rpm
./port install rpm

If everything works as supposed to, a "postinstall" script will also initialize the RPM
DataBase and populate it with all the packages installed by the Port Center along with any
other JRE and JDK installed in your system. If not restricted by the SandBox it will
attempt to process XQuartz also, if installed. These packages will go under the full
control of RPM, you can uninstall them.

The "postinstall" script will also run the "vpkg-provides.sh" script that creates a
virtual package for all the libraries installed at /usr/lib. Further, all Frameworks in
System Library are added as capabilities that the system provides.

If everything has indeed worked as supposed to, you should be able to install easily at
least a small RPM package, ie gnu sed.

Below there are step by step instructions to help you download this small RPM package
(sed-4.2.2-5) from public-yum.oracle.com and install it on an Mac OS Mojave (10.14).


Download a source RPM Package
-----------------------------

cd ~/Downloads
curl -O https://yum.oracle.com/repo/OracleLinux/OL7/latest/x86_64/getPackageSource/sed-4.2.2-5.el7.src.rpm
rpm -i sed-4.2.2-5.el7.src.rpm


The above command is not supposed to complain at all about signature problems, but you
will likely see and it is safe to ignore the following two recurring warnings.

warning: user mockbuild does not exist - using root
warning: group mockbuild does not exist - using root


In case you see a warning about the signature, run:
sudo rpm --import /usr/local/etc/pki/rpm-gpg/RPM-GPG-KEY-oracle-ol7


Porting a source RPM package to Mac OS Mojave (10.14)
-----------------------------------------------------
cd $HOME/rpmbuild
sed -i.sav "s/, [ ]*libselinux-devel//g" SPECS/sed.spec
sed -i.sav "s,/sbin/install-info,/usr/local/bin/install-info,g" SPECS/sed.spec
sed -i.sav "s/configure[ ]*--without-included-regex/configure --disable-nls \
    --disable-i18n --with-included-regex/g" SPECS/sed.spec
sed -i.sav "s/sed.info.gz/sed.info/g" SPECS/sed.spec
sed -i.sav "s/%find_lang/#%find_lang/g" SPECS/sed.spec
sed -i.sav "s/\-f[ ]*%{name}.lang//g" SPECS/sed.spec
rm -f SPECS/sed.spec.sav


Build & Install an RPM Package
------------------------------
cd $HOME/rpmbuild
rpmbuild -ba SPECS/sed.spec --target x86_64
sudo rpm --force --nodeps -i RPMS/X86_64/sed-4.2.2-5.x86_64.rpm


Uninstall an RPM Package
------------------------
Thereafter, one should be able to uninstall this package, which hasn't any active
dependents, with the following command:

sudo rpm --erase sed-4.2.2-5

Since various RPM Scripts depend on gnu sed, you should better reinstall it. If "autogen"
is also installed one can make a complete check on gcc (make check), because the GCC
sub-package "fixincludes" depends on both of them.


Epilogue
--------
Obviously, one can copy paste and execute the above commands in a Mac, as long as Oracle
Linux version 7.6 uses this version of sed. Without any doubt, the RPM installation still
needs fine tuning (no mock, rpmlint, and so on) and probably this package isn't
representative of the effort needed to port a package to OS X. Many RPM Source Packages
require actual patch files to build on a Mac. Once "gnu sed" is installed, one should be
able to install without any modifications "byacc", a prerequisite of "gnu awk", which in
turn requires some path adjustments, ie /sbin and /usr/bin to /usr/local/bin.


Known Issues
------------
[2019-11-26]

- The installation script should run immediately after you clean install macOS Mojave
  along with the Command Line Tools and optionally Xcode. Any other packages on path, ie
  at "/opt/local", might have undesirable side effects; a similar restriction applies to
  Linux as well. This is the only scenario I've successfully tested so far. On Mojave run:

  open /Library/Developer/CommandLineTools/Packages/macOS_SDK_headers_for_macOS_10.14.pkg

- If you rebuild & install an RPM package, check that the following two lines show up:
  "packing     ... <pkg>.rpm"
  "installing  ... <pkg>.rpm"
  If you don't see both lines, then an older RPM is installed (requires manual deletion).

- To install gcc48 or gcc4 on Linux, you have to manually install its dependencies. The
  required dependencies in a RHEL 7.6 system can be satisfied by the official distribution
  packages. So, the PC won't attempt to install ie gettext or pkg-config.

  Although the PlugIn facilitates debugging for Fortran programs with LLDB, one should
  expect that it has bugs (PR/82995 uncovers a F2008 tricky bug). Yet, it can overcome few
  problems mainly on a Mac, like ie the test failure of pr49866.c (PLTOFF isn't acceptable
  by newer mac linkers). On Linux, the PlugIn can compile ie the Fortran program found in
  PR/82065. On May 3, 2020 I couldn't use the PlugIn to build the (static) LAPACK library.

  Both gcc4/gcc48 have a broken "libitm" library on CentOS-7.6 (2020-05-03).

  There are some failures with the "guality" tests on Linux (gcc.dg/quality/) whereas on a
  Mac the PCH tests have been adjusted to run without warnings & core dumps. One of
  the warnings ie didn't comply with a system security policy that loads processes at
  randomly chosen memory addresses; I could avoid it only in the Xcode environment.

  The PCH test "largefile.c" might fail in the gcc48 tests on OS X Yosemite (10.10). An
  extra patch for the PR/14940 has repeatedly bypassed those random failures several times
  success isn't guaranteed though. One can apply manually the patch "gcc48-pr14940.newer"
  to reverse the two related patches (but use then an 1 GB array for the PCH area).

  On Mojave, five tests have been adjusted to run with the option "-fprofile-generate",
  instead of the unsupported option "-pg". Further, during the build you can safely ignore
  any popup that may show up to inform you that the architecture i386 is deprecated.

  The package gcc4 cannot fully recompile all the java classes of gcj and thus this option
  has been deactivated; haven't figured out why, perhaps when java version > 1.7 ?

  One test failure in "libjava" is Darwin specific (ie from 10.9 to 10.14); see PR/48097.
  In this case, one can still create java classes (byte-codes) that run as supposed to.


  Since 2017-11-20, gcc-4.8.5 has experimental support for Fortran SubModules. An
  internal issue is that the submodule separator '@' has been replaced by '$' as three
  test cases (7, 8, and 29) were failing due to assembly errors in both Linux and with the
  PlugIn.

  Since 2020-04-29, gcc-4.8.5 has experimental support for Allocation of Arrays without
  space specification.

  Note: SubModules, Deferred Length Characters, Finalization, and Allocation of Arrays
  without space specification aren't officially implemented in GNU Fortran version "4.8"
  and there are various newer PRs filed in GCC Bugzilla around these features. More
  detailed information about the supported features per version can be found at
  https://gcc.gnu.org/wiki/Fortran2003Status

- A long standing problem of GNU GCC in macOS had been that the destructors of local
  thread objects were running on deallocated memory (Emulated TLS). To my understanding,
  this issue has been solved for the newer Darwin systems with the solution applied to
  PR/78968. I've back-ported this patch to version 4.8 (another patch is applied for the
  PR/58142 if OS X < 10.9).

- The PR/78534 changes the type of character length fields from int (4 bytes) to size_t.
  The latter can be either 4 bytes on 32bit systems or 8 bytes in 64bit systems. Which can
  possibly affect Fortran/C interoperability projects that don't use BIND(C) attributes.

  This patch has been backported in a manner that let's you optionally change the type. By
  default the older GNU Fortran API is preserved (LONG_CHARLEN=0). To change this value in
  an already configured compiler apply the patch "gcc48-pr78534.longer". To configure a
  a fresh compiler with size_t, simply run: "./port configure gcc4  LONG_CHARLEN=1".

  Note that this back-port can change the API but not the actual limit of a string length.

- Too many test cases of Valgrind (3.13) fail on a mac 10.12, ie 73 of the 215 "memcheck"
  tests fail. Whereas the ratio of the failures is worse when one runs all the tests:
  https://bugs.kde.org/show_bug.cgi?id=365327#c23

  The PC script can install Valgrind-3.15 on OS X 10.11 and macOS 10.12-10.14. Support on
  Mojave though is experimental and the installed program needs additional fine tuning.


- If bash is asked to run a non existing command, you may face an unimportant Segmentation
  Fault. This problem which appeared in Sierra remains even with the newer patches applied
  on 2017-10-02. Perhaps, some Linux specific patches should be skipped on Darwin. I have
  not seen this error thereafter in High Sierra (10.13.4).

  To set this bash shell as the user default, one has to manually run this command:
  sudo dscl . create /Users/${LOGNAME} UserShell /usr/local/bin/bash

- The GNU tar-1.26 command (tar) might complain for unknown "header keywords" if the pc
  "tarball" has been compressed by the BSD tar command (bsdtar). I ignore these messages.

  As an interim solution, the PC script can install GNU tar-1.29 on macOS (10.11-10.14).

- The installation script seems to be ready for macOS Mojave (10.14) but it has not
  been tested very well. On 2019-11-30 I've installed gcc (4.8.5 & 7.5), RPM and autogen.

  Below are mentioned two failures I'd faced on 10.13.4. The first of these two failures
  is likely related to the file "include-fixed/stdio.h", which is also found in gcc 7.5:

  - /usr/local/lib/gcc/x86_64-apple-darwin18/4.8.5/include-fixed/stdio.h
  - /opt/local/lib/gcc/x86_64-apple-darwin18/7.5.0/include-fixed/stdio.h

  One can run the following command in the build directory to reproduce the first failure:

  make check-gcc-c RUNTESTFLAGS="cpp.exp=isysroot-1.c -v"

  As an interim solution, the patch "gcc48-fixinc2" makes sure the problematic include
  file is deleted, which happens only when we run "make check-fixincludes", not if you run
  "make stmp-fixinc". Once we have the official solution, this patch will be discarded.

  If you can't place gcc4 to /tmp/gcc.dst, delete this header file from the PC directory:
  rm -rf gcc4/gcc-4.8.5-build/gcc/include-fixed/stdio.h

  The second failure can be reproduced with the following command:

  make check-gcc-c++ RUNTESTFLAGS="dg.exp=darwin-cfstring1.C"

  The applied patch is proposed at: https://gcc.gnu.org/bugzilla/show_bug.cgi?id=83531


- Newer versions of GNU GCC (7.5 & 8.4) are installed at /opt/local instead of /usr/local.
  The PC script creates some soft links (ie /opt/local/bin/gcc -> /opt/local/bin/gcc8)
  without examining if an existing link points to a newer version.

- Note that the Apple JRE is deprecated and some GUI classes have issues after Sierra. In
  specific several text components aren't functional and possibly there are more problems.
  In example, I can't edit BNF Grammars in Syntaxis with the Apple JRE on 10.13 & 10.14.
  Yet, I could fully rebuild parsers/scanners on a fresh installed Mojave and the JAR file
  built by the Apple JDK was fully functional on both Windows-8.1 & RedHat-9 (kernel 2.4).

- You may fail to install Whizard-2.8.2 on Mojave (10.14) due to the following error msg:

   dyld: Symbol not found: __ZTTNSt7__cxx1118basic_stringstreamIcSt11char_traitsIcESaIcEEE

  The above seems to be some recurring problem, it's reported ie also in 2015 and 2017:

    https://groups.google.com/forum/#!topic/fastsimcoal/09fmDduSEEg
    http://mailman.ucar.edu/pipermail/ncl-install/2017-January/002194.html

  So far, I haven't figured out what exactly caused this problem or when introduced. The
  last time I could successfully build the project was likely in December 2019 on a fresh
  installed system (Mojave Installer 10.14.6 - summer 2019). On a MacMini (2011), isolated
  from the Internet, the installation yesterday run smoothly on 10.11-10.13 & CentOS-7.6.

  If you face the above error, then as an interim solution you could try these 2 commands:
    ./port clean    whizard
    ./port install  whizard  whizard-compiler=gcc4

  Although, this problem is fixed since Apr 4, 2020 the proper solution is a clean system.

- If we build & run the following LAPACK test with gfortran we'll likely find differences:
  https://github.com/numericalalgorithmsgroup/LAPACK_Examples

  In two cases (zggev_example.x & zggev3_example.x) we find 1-2 differences with negative
  zeros. Whereas, in another case (zggevx_example.x) there are some numeric differences
  with very small values (exponent < -15).

- You should better upgrade libiconv to 1.16 (./port install libiconv). Since Apr 11, 2020
  this library is patched to provide few dual APIs, useful on Mojave if DYLD_LIBRARY_PATH
  has some values. Otherwise, you may face the error "dyld: Symbol not found: _iconv".


 ------------------------------------------------------------------------------------
 [ The Port Center (pc) was originally hosted at http://users.otenet.gr/~drikosev/ ]
```