PyQ Documentation

Release 3.8.4

Enlightenment Research, LLC.

January 13, 2017

Contents

PyQ provides seamless integration of Python and Q code. It brings Python and Q interpretors in the same process and allows code written in either of the languages to operate on the same data. In PyQ, Python and Q objects live in the same memory space and share the same data.

CHAPTER 1

Quickstart

First, make sure that PyQ is *installed* and *up-to-date*.

Start interactive session:

```
$ pyq
>>> from pyq import q
>>> from datetime import date
```

Create an empty table:

```
>>> q.trade = q('([]date:();sym:();qty:())')
```

Insert sample data:

```
>>> q.insert('trade', (date(2006,10,6), 'IBM', 200))
k(',0')
>>> q.insert('trade', (date(2006,10,6), 'MSFT', 100))
k(',1')
```

Display the result:

```
>>> q.trade.show()
date       sym  qty
-------------------
2006.10.06 IBM  200
2006.10.06 MSFT 100
```

Define a parameterized query:

```
>>> query = q('{[s;d]select from trade where sym=s,date=d}')
```

Run a query:

```
>>> query('IBM', date(2006,10,6))
k('+`date`sym`qty!(,2006.10.06;,`IBM;,200)')
```

Pretty print the result:

```
>>> query('IBM', date(2006,10,6)).show()
date       sym qty
-------------------
2006.10.06 IBM 200
```

CHAPTER 2

Table of Contents

2.1 Installation

2.1.1 OS Support

PyQ has been tested and is supported on Linux and OS X 10.10 or later.

PyQ has support for Solaris, but has not been tested recently.

Windows is not supported yet.

2.1.2 Prerequisites

- kdb+ 3.1 or later;
- Python 2.7, or 3.4 or later;
- GNU make, gcc or clang.

2.1.3 Using pip

Use following pip command to install PyQ into your environment.

```
pip install -i https://pyq.enlnt.com --no-binary pyq pyq
```

You can specify which version you would like to install:

```
pip install -i https://pyq.enlnt.com --no-binary pyq pyq==3.8
```

2.1.4 Using source code

1. Get the source code:
   - You can clone the repository

```
git clone https://github.com/enlnt/pyq.git
```

   - Download the tar file:: or zip file:: and extract it.
2. You can now install into your environment:

```
$ pip install <path to the source>
```

2.2 Keeping PyQ up-to-date

You can upgrade PyQ to the latest version by running:

```
pip install -i https://pyq.enlnt.com --no-binary pyq -U pyq
```

2.3 PyQ Reference

(This section is generated from the PyQ source code. You can access most of this material using pydoc or the built-in help method.)

2.3.1 class pyq.K

2.4 Jupyter Notebook and PyQ

PyQ provides a Jupyter extension for accessing kdb+ from notebook and IPython command line interface.

In Jupyter notebook or the IPython Command Line Interface you can load PyQ's magic.

Start Jupyter notebook:

```
$ pyq -m notebook
```

or, start IPython command line interface:

```
$ pyq -m IPython
```

Then in the cell load pyq magic:

```
%load_ext pyq.magic
```

This gives you access to two magics:

- Line magic:

```
In [2]: %q t:([]a:til 3;b:10*til 3)

In [3]: %q show t
a b
----
0 0
1 10
2 20
```

- Cell magic:

```
In [4]: %%q
   ....: a: exec a from t where b=20
   ....: b: exec b from t where a=2
   ....: a+b
   ....:
Out[4]: ,22
```

You can pass following options to cell magic:

-l (dir|script)
: pre-load database or script

-h host:port
: execute on the given host

-o var
: send output to a variable named var

-i var1, .., varN
: input variables

-1
: redirect stdout

-2
: redirect stderr

2.5 kdb+ Command Line Interface

While in PyQ, you can drop to emulated kdb+ Command Line Interface (CLI). Here is how:

Start pyq:

```
$ pyq
>>> from pyq import q
```

Enter kdb+ CLI:

```
>>> q()
q)t:([]a:til 5; b:10*til 5)
q)t
a b
----
0 0
1 10
2 20
3 30
4 40
```

Exit back to Python:

```
q)\
>>> print("Back to Python")
Back to Python
```

Or you can exit back to shell:

```
q)\\
$
```

2.6 Installing 32-bit PyQ with the free 32-bit kdb+ on 64-bit CentOS 7

2.6.1 1. Install development tools and libraries required to build 32-bit Python

```
$ sudo yum install gcc gcc-c++ rpm-build subversion git zip unzip bzip2 wget
$ sudo yum install libgcc.i686 glibc-devel.i686 glibc.i686 zlib-devel.i686 readline-devel.i686 \
  gdbm-devel.i686 openssl-devel.i686 ncurses-devel.i686 tcl-devel.i686 libdb-devel.i686 bzip2-devel.i
  sqlite-devel.i686 tk-devel.i686 libpcap-devel.i686 xz-devel.i686 libffi-devel.i686
```

2.6.2 2. Download, compile and install the 32-bit version of Python 2.7.12

We are going to install Python 2.7.12 into */opt/python2.i686*.

```
$ mkdir -p ${HOME}/Archive ${HOME}/Build
$ sudo mkdir -p /opt/python2.i686
$ wget http://www.python.org/ftp/python/2.7.12/Python-2.7.12.tgz -O ${HOME}/Archive/Python-2.7.12.tgz
$ tar xzvf ${HOME}/Archive/Python-2.7.12.tgz -C ${HOME}/Build
$ cd ${HOME}/Build/Python-2.7.12
$ export CFLAGS=-m32 LDFLAGS=-m32
$ ./configure --prefix=/opt/python2.i686 --enable-shared | tee c.log
$ LD_RUN_PATH=/opt/python2.i686/lib make | tee m.log
$ sudo make install |tee i.log
$ unset CFLAGS LDFLAGS
```

Let's confirm we've got 32-bit Python on our 64-bit system:

```
$ uname -mip
x86_64 x86_64 x86_64
$ /opt/python2.i686/bin/python -c "import platform; print(platform.processor(), platform.architecture
('x86_64', ('32bit', 'ELF'))
```

Yes, it is exactly what we desired.

2.6.3 3. Install virtualenv into Python installation

We are going to use virtualenv. Let's install it together with pip, setuptools and wheel into Python installation to make things easier.

```
$ sudo /opt/python2.i686/bin/python -mensurepip
$ sudo /opt/python2.i686/bin/python -mpip install -U pip setuptools virtualenv wheel
```

2.6.4 4. Create 32-bit Python virtual environment

Create virtual environment:

```
$ /opt/python2.i686/bin/virtualenv ${HOME}/Work/pyq
```

Define *QHOME* in the new virtuale nvironment:

```
$ echo 'export QHOME=${VIRTUAL_ENV}/q' >> ${HOME}/Work/pyq/bin/activate
```

Enter virtualenvironment we've just created, confirm we've got 32-bit Python in it:

```
$ source ${HOME}/Work/pyq/bin/activate
$ python -c "import struct; print(struct.calcsize('P') * 8)"
32
```

2.6.5 5. Download the 32-bit Linux x86 version of kdb+ from kx.com

Download kdb+ by following this link.

Save downloaded file as *${HOME}/Work/linux-x86.zip*.

2.6.6 6. Extract kdb+ and install PyQ

Extract downloaded file:

```
$ unzip ${HOME}/Work/linux-x86.zip -d ${VIRTUAL_ENV}
```

Install PyQ (note, PyQ 3.8.2 or newer required):

```
$ pip install -i https://pyq.enlnt.com --no-binary pyq pyq>=3.8.2
```

2.6.7 6. Use PyQ

Start PyQ:

```
$ pyq
>>> import platform
>>> platform.processor()
'x86_64'
>>> platform.architecture()
('32bit', 'ELF')
>>> from pyq import q
>>> q.til(10)
k('0 1 2 3 4 5 6 7 8 9')
```

2.7 Version History

2.7.1 PyQ 3.8.4

Released on 2017-01-13

- !414 - #843: Setup should not fail if VIRTUAL_ENV is undefined
- !395 - #825: Fixed uninitialized “readonly” field in getbuffer

2.7.2 PyQ 3.8.3

Released on 2016-12-15

- !357 - #799: Several documentation fixes.
- !368 - #802: Setup should not fail if $VIRTUAL_ENV/q does not exist.

2.7.3 PyQ 3.8.2

Released on 2016-12-01

Documentation improvements:

- !306 - #763: Update README.md - fixed INSTALL link.
- !312 - Fix formatting; ?? -> date of the release in the CHANGELOG.
- !322 - Fixed formatting error in the documentation.
- !324 - #744: use pip to install from the source.
- !338 - #785: Virtual environment setup guide.
- !346 - #764: docs improvements
- !342 - #787: Added links to rtd documentation.

PyQ executable improvements:

- !310 - #761: Allow PyQ executable to be compiled as 32-bit on 64-bit platform.
- !329 - #646: Print PyQ, KDB+ and Python versions if –versions option is given to pyq.
- !332 - #646: Print full PyQ version.
- !333 - #781: Find QHOME when q is installed next to bin/pyq but no venv is set.
- !336 - #783: Fixed a bug in CPUS processing
- !345 - #646: Added NumPy version to –versions output.

Other improvements and bug fixes:

- !308 - #759: Return an empty slice when (stop - start) // stride < 0.
- !320 - #771: Workaround for OrderedDict bug in Python 3.5
- !323 - #773: Renamed ipython into jupyter; added starting notebook command.
- !326 - #720: Simplified the test demonstrating the difference in Python 2 and 3 behaviors.
- !327 - #720: Finalize embedded Python interpreter on exit from q.
- !331, !343 - #768: Improve C coverage

Improvement in the (internal) CI:

- !305, !309, !311, !321, !335, !347 - Multiple improvements in the CI.
- !319 - #770: Run doctests in tox.

2.7.4 PyQ 3.8.1

Released on 2016-06-21

- !292 - #744: Print guessed path of q executable when exec fails.
- !293, !294 - #748 Use VIRTUAL_ENV environment variable to guess QHOME.
- !301, !295 - #751: Update documentation.
- !296 - #750: Fall back on 32-bit version of q if 64-bit version does not run.
- !298, !299, !300, !303 - #753: CI Improvements.

- !302 - #755: Use preserveEnumerations=1 option to b9 instead of -1.

2.7.5 PyQ 3.8

Released on 2016-04-26.

- !256 - #670: Enable 32-bit CI
- !258 - #717 Expose sd0 and sd1 in python.
- !259 - #718 Added a test running "q test.p".
- !261 - Use Python 3.4.3 in CI
- !272, !273 - #731 Added Python 3.5.0 test environment and other CI improvements.
- !263 - #718 More p) tests
- !264 - #709 Redirect stderr and stdout to notebook
- !271 - #729 Conversion of lists of long integers to q.
- !274 - #728 Don't corrupt existing QHOME while running tox.
- !275 - #733 Don't add second soabi for Python 3.5.
- !276 - #734: Added support for enums in memoryview.
- !277 - #736: Implemented format() for more scalar types.
- !278 - #737 Misleading error message from the list of floats conversion.
- !279, !280 - #738 CI improvements
- !281 - #611: Updated k.h as of 2016.02.18
- !286, !288, !289, !290 - #742 PyQ Documentation
- !287 - #745: Automatically generate version.py for PyQ during setup.

2.7.6 PyQ 3.7.2

Released on 2015-07-28.

- !270 - #726 Reuse dict converter for OrderedDict.
- !267 - #724 and #723 numpy <> q conversion fixes.
- !266 - #725 Use 001..002 to bracket ANSI escapes.
- !265 - #721 Made slicing work properly with associations (dictionaries) and keyed tables.
- !260 - #719 Backport python 3 bug fixes.
- CI Improvements (!257, !262, !269, !268).

2.7.7 PyQ 3.7.1

Released on 2015-02-12.

- !244 - #701 Fixed using q datetime (z) objects in format().
- !246 - Removed pytest-pyq code. pytest-pyq is now separate package.

- !247 - #709 IPython q-magic improvements
- !248 - #673 Implemented unicode to q symbol conversion in python 2.x.
- !249, !252 - #691 Improved test coverage
- !250, !251 - #695 Use Tox as test-runner
- !253 - #715 Fixed table size computation in getitem.
- !255 - #691 Remove redundant code in slice implementation

2.7.8 PyQ 3.7

Released on 2015-01-15.

- !222 - #581 Implements conversion of record arrays.
- !223 - #680 Fixed int32 conversion bug.
- !224 - #681 Fixed datetime bug - freed memory access.
- !225 - Added support for numpy.int8 conversion.
- !226 - #644 Fixed descriptor protocol.
- !227 - #663 Fixed nil repr (again).
- !228, !233, !237, !239 - #687 Updates to documentation in preparation to public release.
- !229 - #690 Use only major kx version in _k module name.
- !230 - #691 Added tests, fixed date/time list conversion.
- !232 - #693 Implement pyq.magic.
- !234 - #694 Use single source for python 2 and 3. (No 2to3.)
- !235 - #674 Added support for nested lists.
- !236 - #678 Fixed compiler warnings.
- !238 - #657 Make numpy optional.
- !240 - #674 Added support for nested tuples.
- !241 - #696 Implemented slicing of K objects.
- !242 - #699 int and float of non-scalar will raise TypeError.
- !243 - #697 Fixed a datatime bug.

2.7.9 PyQ 3.6.2

Released on 2014-12-23.

- !198 - #654 Restore python 3 compatibility
- !211 - #667 Added pyq.c into MANIFEST
- !213 - #669 Fixed a crash on Mac
- !214 - #590 Implemented numpy date (M8) to q conversion
- !215, !216 - #590 Implemented support for Y, M, W, and D date units
- !217, !218, !220, !221 - #666 Multiple CI improvements

- !219 - #676 Implemented numpy.timedelta64 to q conversion

2.7.10 PyQ 3.6.1

Released on 2014-11-06.

- !206 - #663 Fixed nil repr
- !207 - CI should use cached version of packages
- !208 - #665 Allow K objects to be written into ipython zmq.iostream
- !209 - Show python code coverage in CI
- !210 - #666: Extract C and Python coverage to print in the bottom of the CI run
- !212 - Bump version to 3.6.1b1

2.7.11 PyQ 3.6.0

Released on 2014-10-23.

- !189 - #647 Fix pyq.q() prompt
- !190 - CI should use Python 2.7.8
- !191 - #648 Boolean from empty symbol should be False
- !192 - #634: Moved time converter to C and removed unused converters
- !193 - #652 Added __long__ method to K type.
- !194 - #653 Allow K integer scalars to be used as indices
- !195, !197 - #651 Format for scalar types D, M, T, U, and V.
- !196 - #611 Updated k.h to 2014.09.11
- !199 - #656 Iteration over K scalars will now raise TypeError.
- !200 - #655 Added support for Python 3 in CI
- !202 - #571 Added support for uninstalling Q components
- !203 - #633 Improve test coverage
- !204 - #633 Added boundary and None checks in ja

2.7.12 PyQ 3.5.2

Released on 2014-07-03.

- !184, !186 - #639 taskset support. Use CPUS variable to assign CPU affinity.
- !187 - #641 color prompt
- !185 - #640 Restore minimal support for old buffer protocol

2.7.13 PyQ 3.5.1

Released on 2014-06-27.

- !177, !178 – #631 pyq is binary executable, not script and can be used in hasbang.
- !179 – #633 Added memoryview tests.
- !181 – #636 Moved extension module into pyq package.
- !182 – #633 Removed old buffer protocol support.
- !183 - #638 Calling q() with no arguments produces an emulation of q) prompt

2.7.14 PyQ 3.5.0

Released on 2014-06-20.

- !164 – #611 Updated k.h
- !165 – #614 Expose jv
- !166 – #580 Show with output=str will return string
- !167 – #627 Fixed p language
- !168 – Fix for pip, pycharm and OS X
- !169 – #629 python.py script was renamed to pyq
- !170 – #632 jv reference leak
- !171 – #633 C code review
- !172 – #634 k new
- !173 – #612 Generate C code coverage for CI
- !174, !175 – #633 test coverage
- !176 – #635 Disable strict aliasing

2.7.15 PyQ 3.4.5

Released on 2014-05-27.

- 614: Expose dj and ktj
- 620: Empty table should be falsy
- 622: Convert datetime to “p”, not “z”

2.7.16 PyQ 3.4.4

Released on 2014-05-23.

- python.q returns correct exit code

2.7.17 PyQ 3.4.3

Released on 2014-04-11.

- 617: Dict Conversion
- 619: Len Keyed Table

2.7.18 PyQ 3.4.2

Released on 2014-04-11.

- 589: Symbol array roundtripping
- 592: Properly register py.path.local
- 594: Support passing additional valuse to select/update/exec methods.
- 595: Implement pytest_pyq plugin
- 596: Implement python dict converter
- 601: Add support for ^ (fill) operator
- 602: Fix r-ops for non-commutative operations.
- 603: Fix unary + and implement unary ~
- 604: Make all q methods accessible from pyq as attributes
- 609: Updated k.h to the latest kx version
- NUC: Only true division is supported. Use "from __future__ import division" in python 2.x.

2.7.19 PyQ 3.4.1

Released on 2014-03-14.

- Add support for char arrays #588
- PyQ can now be properly installed with pip -r requirements.txt #572

2.7.20 PyQ 3.4

Released on 2014-03-07.

- Issues fixed: #582, #583, #584, #586
- Support dictionary/namespace access by .key
- Support ma.array(x) explicit conversion
- Add support for comparison of q scalars

2.7.21 PyQ 3.3

Released on 2014-02-05.

- Issues fixed: #574, #575, #576, #577, #578

2.7.22 PyQ 3.2

Released on 2013-12-24.

- Issues fixed: #556, #559, #560, #561, #562, #564, #565, #566, #569, #570, #573
- **NEW: wrapper for python.q to use it under PyCharm** Note: You will need to create symlink from python to python.py in order for this to work, ie: ln -s bin/python.py bin/python
- Support to use 32-bit Q under 64-bit OS X

2.7.23 PyQ 3.2.0 beta

- Convert int to KI if KXVER < 3, KJ otherwise
- In Python 2.x convert long to KJ for any KXVER

2.7.24 PyQ 3.1.0

Released on 2012-08-25.

- support Python 3.2
- release pyq-3.1.0 as a source archive

2.7.25 2012-08-10

- basic guid support

2.7.26 PyQ 3.0.1

Released on 2012-08-09.

- support both q 2.x and 3.x
- better setup.py
- release pyq-3.0.1 as a source archive

2.7.27 2009-10-23

- NUC: k3i
- K(None) => k("::")
- K(timedelta) => timespan

2.7.28 2009-01-02

- Use k(0, ..) instead of dot() and aN() to improve compatibility
- Default to python 2.6
- Improvements to q script.p
- NUC: extra info on q errors

2.7.29 2007-03-30

implemented K._ja

2.7.30 0.3

- Added support for arrays of strings

2.7.31 0.2

- Implemented iterator protocol.

2.8 PyQ General License

2.8.1 Copyright

2.8.2 Free 32-bit license

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

0. This software or its derivatives must be run on a free 32-bit version of kdb+ subject to the DOWNLOAD KDB+ SOFTWARE LICENSE AGREEMENT.
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the Enlightenment Research, LLC.
4. Neither the name of the Enlightenment Research, LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY ENLIGHTENMENT RESEARCH, LLC ''AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ENLIGHTENMENT RESEARCH, LLC BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

2.8.3 Commercial license

For a 64-bit license and support, please contact “PyQ License” pyq-lic@enlnt.com.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search