
PvMail Documentation

Release 3.0

Pete Jemian, APS, ANL <jemian@anl.gov>

November 28, 2011

CONTENTS

1	Contents	3
1.1	Contents	3
2	Glossary	13
3	Dependencies	15
	Python Module Index	17
	Index	19

PvMail was built to watch (monitor) an EPICS PV and send an email when the value of that PV changes from 0 to 1.

The PV being watched (that *triggers* the sending of the email) can be any EPICS record type or field that results in a value of 0 (zero) that changes to 1 (one). This includes these EPICS records (and possibly more): *ai, ao, bi, bo, calcout, scalcout, swait*

When an event cause an email to be triggered, PvMail will retrieve the value of another PV that is the first part of the message to be sent. Additional metadata will be appended to the message.

Note: Email is sent using a call to the `mail` program on the native OS. This almost certainly precludes its use on Windows systems. The GUI or command-line versions will operate but likely no email is sent. Also, the host computer must allow sending email to the intended recipients.

PvMail provides either a command-line interface or a graphical user interface. Both are accessed from the same command, using different command-line options. The command-line version is intended to run as a background program, it has no user interaction but logs all its output into a log file. The GUI version provides a screen to edit each of the parameters before the background process is started. It also provides buttons to start and stop the background process.

CONTENTS

1.1 Contents

1.1.1 pvMail.py: command-line interface

PvMail is started from the command line:

```
$ ./pvMail.py pvMail:trigger pvMail:message jemian
```

No program output is printed to the screen. Instead, the output is directed to a log file. Here is an example:

```
INFO:root:(pvMail.py,2011-11-27 19:03:23.072392) #####
INFO:root:(pvMail.py,2011-11-27 19:03:23.072826) startup
INFO:root:(pvMail.py,2011-11-27 19:03:23.072897) trigger PV      = pvMail:trigger
INFO:root:(pvMail.py,2011-11-27 19:03:23.073323) message PV     = pvMail:message
INFO:root:(pvMail.py,2011-11-27 19:03:23.073401) email list      = ['jemian']
INFO:root:(pvMail.py,2011-11-27 19:03:23.073463) log file        = logfile.log
INFO:root:(pvMail.py,2011-11-27 19:03:23.073553) message file     = pvmail_email.txt
INFO:root:(pvMail.py,2011-11-27 19:03:23.073667) logging interval = 5.0
INFO:root:(pvMail.py,2011-11-27 19:03:23.073735) sleep duration  = 0.2
INFO:root:(pvMail.py,2011-11-27 19:03:23.073795) interface       = command-line
INFO:root:(pvMail.py,2011-11-27 19:03:23.073855) user           = jemian
INFO:root:(pvMail.py,2011-11-27 19:03:23.073952) host           = como-ubuntu64
INFO:root:(pvMail.py,2011-11-27 19:03:23.074053) program        = ./pvMail.py
INFO:root:(pvMail.py,2011-11-27 19:03:23.074124) PID             = 8903
INFO:root:(pvMail.py,2011-11-27 19:03:23.074196) do_start
INFO:root:(pvMail.py,2011-11-27 19:03:23.074280) test connect with pvMail:message
INFO:root:(pvMail.py,2011-11-27 19:03:23.445334) test connect with pvMail:trigger
INFO:root:(pvMail.py,2011-11-27 19:03:23.468540) passed basicChecks(), starting monitors
INFO:root:(pvMail.py,2011-11-27 19:03:23.477917) checkpoint
INFO:root:(pvMail.py,2011-11-27 19:03:27.373142) pvMail:trigger = 1
INFO:root:(pvMail.py,2011-11-27 19:03:27.373908) SendMessage
INFO:root:(pvMail.py,2011-11-27 19:03:27.374199) sending email to: jemian
INFO:root:(pvMail.py,2011-11-27 19:03:27.374716) mail -s "pvMail.py: pvMail:trigger" jemian < /tmp/pv
INFO:root:(pvMail.py,2011-11-27 19:03:27.538022) message(s) sent
INFO:root:(pvMail.py,2011-11-27 19:03:28.092551) checkpoint
INFO:root:(pvMail.py,2011-11-27 19:03:29.440516) pvMail:trigger = 0
```

The program starts, reports its configurations, and connects with the EPICS PVs, and then goes into a background mode. A checkpoint (command-line option `-i`) is reported periodically. The default is 5 seconds. This may be changed to 10 minutes or longer for production use, but is always specified in seconds.

Observe that, in the above example, the trigger PV changed from 0 to 1 at 19:03:27.373142 (and back to 0 at 19:03:29.440516). The change at ~19:03:27 triggered PvMail to send an email as configured. For now, the code

writes the text of the email to a temporary file (command-line option `-m`, default is `"/tmp/pvmail_message.txt"`). In this example, the message reads:

```
pvMail default message
```

```
user: jemian
host: como-ubuntu64
date: 2011-11-27 19:03:27.374135
program: ./pvMail.py
PID: 8903
trigger PV: pvMail:trigger
message PV: pvMail:message
recipients: jemian
```

The message shows up in the mail browser (here my Linux mail program):

```
jemian@como-ubuntu64$ mail
Mail version 8.1.2 01/15/2001.  Type ? for help.
"/var/mail/jemian": 3 messages 3 new
>N 1 jemian@como-ubunt  Sun Nov 27 18:27  25/730  pvMail.py: pvMail:trigger
  N 2 jemian@como-ubunt  Sun Nov 27 18:58  25/730  pvMail.py: pvMail:trigger
  N 3 jemian@como-ubunt  Sun Nov 27 19:03  25/730  pvMail.py: pvMail:trigger
```

The full message, as seen in the mail browser is:

```
Message 3:
From jemian@como-ubuntu64 Sun Nov 27 19:03:27 2011
Envelope-to: jemian@como-ubuntu64
Delivery-date: Sun, 27 Nov 2011 19:03:27 -0600
To: jemian@como-ubuntu64
Subject: pvMail.py: pvMail:trigger
From: Pete R Jemian <jemian@como-ubuntu64>
Date: Sun, 27 Nov 2011 19:03:27 -0600
```

```
pvMail default message
```

```
user: jemian
host: como-ubuntu64
date: 2011-11-27 19:03:27.374135
program: ./pvMail.py
PID: 8903
trigger PV: pvMail:trigger
message PV: pvMail:message
recipients: jemian
```

usage

When PvMail is started from the command line with no additional parameters:

```
$ pvMail.py

usage: pvMail.py [-h] [-l LOG_FILE] [-m MESSAGE_FILE] [-i LOGGING_INTERVAL]
                [-r SLEEP_DURATION] [-g] [-v]
                trigger_PV message_PV email_addresses
pvMail.py: error: too few arguments
```

This is the *usage* message and it tells us that we must always supply three additional items: `trigger_PV` `message_PV` `email_addresses`.

help

It may be easier to view the help:

```
$ ./pvMail.py --help
usage: pvMail.py [-h] [-l LOG_FILE] [-m MESSAGE_FILE] [-i LOGGING_INTERVAL]
               [-r SLEEP_DURATION] [-g] [-v]
               trigger_PV message_PV email_addresses
```

Watch an EPICS PV. Send email when it changes from 0 to 1.

positional arguments:

```
trigger_PV          EPICS trigger PV name
message_PV          EPICS message PV name
email_addresses     email address(es), comma-separated if more than one
```

optional arguments:

```
-h, --help          show this help message and exit
-l LOG_FILE         for logging program progress and comments
-m MESSAGE_FILE     temporary file for email message
-i LOGGING_INTERVAL checkpoint reporting interval (s) in log file
-r SLEEP_DURATION  sleep duration (s) in main event loop
-g, --gui          Use the graphical rather than command-line interface
-v, --version      show program's version number and exit
```

1.1.2 pvMail.py: graphical user interface

The PvMail program GUI is started from the command line with the `-g` or `--gui` command-line options. Without either of these, the command-line interface is started.:

```
$ ./pvMail.py pvMail:trigger pvMail:message jemian -g
```

-tba-

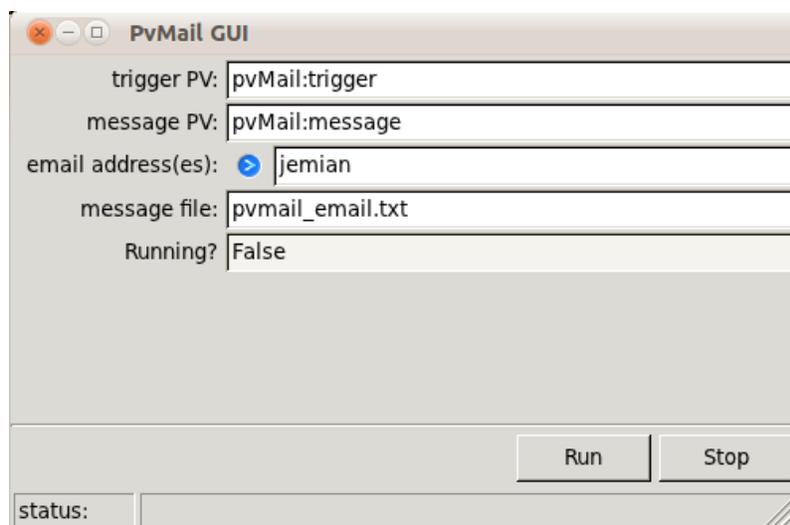


Figure 1.1: GUI of the *PvMail* application

1.1.3 EPICS test database

To test the program during its development, a test database (*test.db*) was prepared. The database creates two PVs:

pvMail:trigger the PV to watch

pvMail:message the message to be sent

starting: softIoc

Start the database by adding it to an existing EPIC IOC configuration or by starting a soft IOC using the `softIoc` program `softIOC` from EPICS base. Here is an example of how that looks from a Linux command shell:

```
1 $ softIoc -d test.db
2 Starting iocInit
3 #####
4 ## EPICS R3.14.12 $Date: Wed 2010-11-24 14:50:38 -0600$
5 ## EPICS Base built Feb 27 2011
6 #####
7 iocRun: All initialization complete
8 epics>
```

Note: Here, the shell prompt is signified by the `$` symbol.

watching: camonitor

Once the EPICS IOC is started and the PVs are available, it is possible to watch them for any changes from the command line using the `camonitor` application from EPICS base:

```
$ camonitor pvMail:trigger pvMail:message
  pvMail:trigger          <undefined> off UDF INVALID
  pvMail:message         <undefined> pvMail default message UDF INVALID
```

Note: Do not be concerned about the `UDF INVALID` notices, they will disappear once the PVs have been written to at least once.

changing a PV: caput

You can test changing the value of the trigger PV using the `caput` application from EPICS base:

```
$ caput pvMail:trigger 1
  Old : pvMail:trigger          off
  New : pvMail:trigger          on
```

test.db

Here is the full listing of the test EPICS database used for program development.

```
1 ##### SVN repository information #####
2 # $Date: 2011-11-25 20:16:30 -0600 (Fri, 25 Nov 2011) $
3 # $Author: jemian $
```

```

4 # $Revision: 659 $
5 # $URL: https://subversion.xor.aps.anl.gov/bcdaext/pvMail/src/PvMail/test.db $
6 # $Id: test.db 659 2011-11-26 02:16:30Z jemian $
7 ##### SVN repository information #####
8
9 # EPICS database to use while testing and developing pvMail.py code
10
11 # /APShare/epics/base-3.14.12.1/bin/linux-x86-el5-debug/softIoc -d test.db
12 #
13 # IOC:      softIoc -d test.db
14 # client:  camonitor pvMail:{trigger,message}
15 # pvMail:  pvMail.py pvMail:trigger pvMail:message prjemian@gmail.com, jemian@anl.gov
16
17 record(bo, "pvMail:trigger")
18 {
19     field(DESC, "trigger PV")
20     field(ZNAM, "off")
21     field(ONAM, "on")
22 }
23 record(stringout, "pvMail:message")
24 {
25     field(DESC, "message to be sent by email")
26     field(VAL, "pvMail default message")
27 }

```

1.1.4 PvMail as a Python package

This section provides the source code documentation. The documentation here may be of little or no use to the casual user of this software.

installation

The PvMail project can be installed as a Python package.

1. Checkout the project from subversion
2. Change into the project working directory
3. Run `setup.py install`

starter program

Once the PvMail project has been installed as a package, the PvMail application can be run using a simple Python script (included in the project tree at the top-level directory as `pvMail.py`). Here is the current version of that script.

```

1  #!/usr/bin/env python
2
3  '''
4  runs the PvMail application
5
6  see the help option for more details::
7
8      pvMail.py --help
9  '''
10

```

```
11 from PvMail import pvMail
12 pvMail.main()
```

1.1.5 PvMail source code documentation

PvMail Python Package

Source code documentation for EPICS PvMail.

pvMail Module

pvMail: combined CLI and GUI Functionally based on pvMail UNIX shell script written in 1999.

Summary Watches an EPICS PV and sends email when it changes from 0 to 1. PV value can be either integer or float.

Note: When “running”, wait for trigger PV to go to 1. When that happens, fetch mail message from message PV. Then, send that message out to each of the email addresses. The message content should be prioritized for view on a pager or a PDA or smartphone.

author Kurt Goetze

author Pete Jemian

organization AES/BCDA, Advanced Photon Source, Argonne National Laboratory

license Copyright (c) 2011, UChicago Argonne, LLC Copyright (c) 2009, The University of Chicago, as Operator of Argonne National Laboratory. Copyright (c) 2009 The Regents of the University of California, as Operator of Los Alamos National Laboratory. This file is distributed subject to a Software License Agreement found in the file LICENSE that is included with this distribution.

note Version History: 05.09.07 kag Initial alpha version. Needs testing. 2009-12-02 prj: converted to use wxPython (no Tkinter or Pmw) 2011-11-23 prj: complete rewrite using PyEpics and combined GUI (Traits) and CLI

requires EPICS system (<http://www.aps.anl.gov/epics>) with at least two process variables (PVs) where the “Trigger PV” toggles between values of 0 and 1 and the “SendMessage PV” contains a string to send as part of the email message.

class PvMail.pvMail.PvMail

Bases: threading.Thread

Watch an EPICS PV (using PyEpics interface) and send an email when the PV changes from 0 to 1.

basicChecks ()

check for valid inputs, raise exceptions as discovered, otherwise no return result

do_restart ()

restart watching for triggers

do_start ()

start watching for triggers

do_stop ()

stop watching for triggers

receiveMessageMonitor (*value*, ***kw*)
 respond to EPICS CA monitors on message PV

receiveTriggerMonitor (*value*, ***kw*)
 respond to EPICS CA monitors on trigger PV

send_test_message ()
 sends a test message, used for development only

testConnect (*pvname*, *timeout=5.0*)
 create PV, wait for connection, return connection state (True | False)
 adapted from PyEpics `__createPV()` method

class PvMail.pvMail.**SendMessage** (*pvm*)
 Bases: `threading.Thread`
 initiate sending the message in a separate thread

PvMail.pvMail.**basicMailTest** ()
 simple test sending mail using the PvMail class

PvMail.pvMail.**basicStartTest** ()
 simple test of the PvMail class

PvMail.pvMail.**cli** (*results*)
 command-line interface to the PvMail class

Parameters **results** (*obj*) – default parameters from argparse, see main()

PvMail.pvMail.**gui** (*results*)
 graphical user interface to the PvMail class

Parameters **results** (*obj*) – default parameters from argparse, see main()

PvMail.pvMail.**logger** (*message*)
 log a report from this class.

Parameters **message** (*str*) – words to be logged

PvMail.pvMail.**main** ()
 parse command-line arguments and choose which interface to use

PvMail.pvMail.**sendMail** (*subject*, *message*, *recipients*)
 send an email message

Parameters

- **subject** (*str*) – short text for email subject
- **message** (*str*) – full text of email body
- **recipients** (*[str]*) – list of email addresses to receive the message

traits_gui Module

pvMail: just the GUI Build the Graphical User Interface for PvMail using the Traits library from the Enthought Python Distribution.

Copyright (c) 2011, UChicago Argonne, LLC

class PvMail.traits_gui.**ActionHandler**
 Bases: `traitsui.handler.Handler`
 implements controls for PvMail GUI application

do_run (*uinfo*)

start watching the EPICS triggerPV

Parameters **uinfo** (*obj*) – UIInfo object passed from the Action()

Traits Handler method that responds to a Traits Action()

do_stop (*uinfo*)

stop watching the EPICS triggerPV

Parameters **uinfo** (*obj*) – UIInfo object passed from the Action()

Traits Handler method that responds to a Traits Action()

class PvMail.traits_gui.**PvMail_GUI** (*triggerPV='', messagePV='', recipients=['', ''], message_file='gui.log', **kwtraits*)

Bases: traits.has_traits.HasTraits

GUI used for pvMail, declared using Enthought's Traits module

SetStatus (*msg*)

put text in the status box

actionRun = <traitsui.menu.Action object at 0x4e79e30>

actionStop = <traitsui.menu.Action object at 0x4e79e90>

1.1.6 More Information

subversion repository

The PvMail project is hosted on the APS XSD subversion server. You may check out the entire project source code subversion repository and development subdirectory:

```
svn co https://subversion.xor.aps.anl.gov/bcdaext/pvMail ./pvMail
```

project management site

You may find it easier to view the various code revisions and other aspects of the project from the project management site. A link there will direct you to the resources (documentation, source code) for the PvMail project.

<https://subversion.xor.aps.anl.gov/trac/bcdaext>

Documentation

Documentation for the PvMail project, maintained using sphinx (<http://sphinx.pocoo.org>), can be accessed from the WWW at <https://subversion.xor.aps.anl.gov/bcdaext/pvMail/docs/build/html/index.html> PvMail project and also EPUB epub and PDF PDF versions.

1.1.7 TODO items for the next release

1. Complete the documentation of command-line options
2. Complete the GUI documentation
3. Connect status updates from `pVMail.PvMail()` with status in the GUI
4. Report PV connection problems in an obvious way

1.1.8 License

Copyright (c) 2009-2012, UChicago Argonne, LLC

All Rights Reserved

pvMail

BCDA, Advanced Photon Source, Argonne National Laboratory

OPEN SOURCE LICENSE

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Software changes, modifications, or derivative works, should be noted with comments and the author and organization's name.
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. Neither the names of UChicago Argonne, LLC or the Department of Energy nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
4. The software and the end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software produced by UChicago Argonne, LLC under Contract No. DE-AC02-06CH11357 with the Department of Energy."

DISCLAIMER

THE SOFTWARE IS SUPPLIED "AS IS" WITHOUT WARRANTY OF ANY KIND.

Neither the United States GOVERNMENT, nor the United States Department of Energy, NOR uchicago argonne, LLC, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, data, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

GLOSSARY

CA EPICS Channel Access protocol

CLI command-line interface

EPICS <http://www.aps.anl.gov/epics>

GUI graphical user interface

IOC EPICS Input/Output Controller, the EPICS server

message PV EPICS PV that provides the text to be sent by email, additional metadata is appended to this text

OS operating system

PV EPICS process variable

trigger PV EPICS PV that signals an email is to be sent

DEPENDENCIES

This software was built with various standard Python packages available in Python 2.7. Additionally, this program uses:

PyEpics (EPICS interface) <http://cars9.uchicago.edu/software/python/pyepics3/>

Traits (GUI library) <http://code.enthought.com/projects/traits/>

Both of these are available for *easy_install* from the Python Package Index (<http://pypi.python.org/pypi>).

PYTHON MODULE INDEX

p

PvMail.pvMail, 8

PvMail.traits_gui, 9

INDEX

A

ActionHandler (class in PvMail.traits_gui), 9
actionRun (PvMail.traits_gui.PvMail_GUI attribute), 10
actionStop (PvMail.traits_gui.PvMail_GUI attribute), 10

B

basicChecks() (PvMail.pvMail.PvMail method), 8
basicMailTest() (in module PvMail.pvMail), 9
basicStartTest() (in module PvMail.pvMail), 9

C

CA, 13
camonitor, 6
caput, 6
CLI, 13
cli() (in module PvMail.pvMail), 9

D

do_restart() (PvMail.pvMail.PvMail method), 8
do_run() (PvMail.traits_gui.ActionHandler method), 9
do_start() (PvMail.pvMail.PvMail method), 8
do_stop() (PvMail.pvMail.PvMail method), 8
do_stop() (PvMail.traits_gui.ActionHandler method), 10

E

EPICS, 13
epub, 10

G

GUI, 13
gui() (in module PvMail.pvMail), 9

I

IOC, 13

L

logger() (in module PvMail.pvMail), 9

M

main() (in module PvMail.pvMail), 9

message PV, 13

O

OS, 13

P

PDF, 10
PV, 13
PvMail (class in PvMail.pvMail), 8
PvMail project, 10
PvMail.pvMail (module), 8
PvMail.traits_gui (module), 9
PvMail_GUI (class in PvMail.traits_gui), 10
PyEpics, 15

R

receiveMessageMonitor() (PvMail.pvMail.PvMail method), 8
receiveTriggerMonitor() (PvMail.pvMail.PvMail method), 9

S

send_test_message() (PvMail.pvMail.PvMail method), 9
sendMail() (in module PvMail.pvMail), 9
SendMessage (class in PvMail.pvMail), 9
SetStatus() (PvMail.traits_gui.PvMail_GUI method), 10
softIOC, 6
subversion repository, 10

T

test.db, 6
testConnect() (PvMail.pvMail.PvMail method), 9
TODO items, 10
Traits, 15
trigger PV, 13