
DAQOperator 操作用

Python コマンドマニュアル

DAQ Operator を python スクリプトで動作させるために

初版 : 2010 年 7 月 2 日
改訂版 : 2012 年 4 月 9 日

目次

概要	2
DAQ ミドルウェアを動作させるコマンドとして利用する方法	3
Python スクリプトの中で利用する方法	4
__init__ constructor	4
定義	4
説明	4
setURLBase method	4
定義	4
説明	4
configure method	5
定義	5
説明	5
start method	5
定義	5
説明	5
stop method	5
定義	5
説明	5

pause method.....	5
定義.....	5
説明.....	6
resume method.....	6
定義.....	6
説明.....	6
unconfigure method.....	6
定義.....	6
説明.....	6
getLog method.....	6
定義.....	6
説明.....	6
setRunNumber method.....	7
定義.....	7
説明.....	7
cli method.....	7
定義.....	7
説明.....	7
例題.....	7
参考文献.....	9

概要

DAQ ミドルウェアを動作させるため、WEB を用いた GUI である DAQ オペレーションパネルを利用する(文献 1、2、3)。しかし簡単な自動化を図りたい場合、スクリプトを使って DAQ ミドルウェアを動作させることができると便利である。

このマニュアルでは Python 言語で書かれた `daqmwcom` クラスを解説する。このクラスは DAQ オペレータに `configure` コマンドや `start` コマンドなどのコマンドを送るために作られた。利用の方法には 2 通りある。1 つは、DAQ ミドルウェアを動作させるためのコマンドとして動作させる方法である。もう 1 つの方法は、このクラスを `import` して Python スクリプトの中で利用する方法である。

Scientific Linux 5 以前の場合と、Scientific Linux 6 以後の場合とでは、URL の指定において異なっていることに注意。

DAQ ミドルウェアを動作させるコマンドとして利用する方法

コマンドの一覧は下記のとおりである。

```
% /usr/bin/daqcom urlbase options
```

urlbase:

<http://localhost/daqmw/operatorPanel/> for SL(C)5 or earlier

or,

<http://localhost/daqmw/scripts/> for SL(C)6 or later

options:

-c or --configure : configure command

-b runNum or --start runNum : start(begin) command

for example, -b 100

-e or --stop : stop(end) command

-u or --unconfigure : unconfigure command

-p or --pause : pause command

-r or --resume : resume command

-g tag or --getLog tag : getLog command

for example, -g state or -g all

all means all of tags

start コマンドはラン番号を付ける必要がある。この番号は **Web** を利用した場合と同じように、記録されて保存される。**getLog** コマンドは **DAQ** オペレータから **Log** 情報を取り出すコマンドだが、**tag** を入れることで必要な情報のみを取得できる。例えば、**state** と入れると **state** 情報。**status** と入れると **status** 情報。**eventNum** はイベント情報。**all** とするとすべての情報が入る。**all** 情報は例えば下記の通り。

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet href="style.xsl" type="text/xsl" ?>
<response>
<methodName>Log</methodName>
<returnValue>
<result>
<status>OK</status>
<code>0</code>
<className/><name/>
<methodName/>
<messageEng/>
<messageJpn/>
</result>
<logs>
<log>
<compName>group0:EchoMonitor0</compName>
<state>CONFIGURED</state>
<eventNum>15500</eventNum>
```

```
<compStatus>WORKING</compStatus>
</log>
<log>
<compName>group0:EchoReader0</compName>
<state>CONFIGURED</state>
<eventNum>15500</eventNum>
<compStatus>WORKING</compStatus>
</log>
</logs>
</returnValue>
</response>
```

Python スクリプトの中で利用する方法

スクリプトから利用するため、Python クラスを構成する。

`__init__` constructor

定義

```
def __init__(self, urlbase=None):
```

説明

これは `daqmwcom` のコンストラクタで、まずはこのクラスを `import` する。
`from daqmwcom import daqmwcom`
2通りの呼び出し方がある。1つは、
`com = daqmwcom(url)`
引数付きの呼び出しはベースとなる URL を `url` で設定する。もう1つは、
`com = daqmwcom()`
引数なしの呼び出しはベースとなる URL を後で `setURLBase()` を使って設定する必要がある。

`setURLBase` method

定義

```
def setURLBase(self, urlbase):
```

説明

これは例えば次のように呼び出す。
`com.setURLBase("http://localhost/daqmw/operatorPanel/")`

ベースとなる URL を指定してその後のメソッドのアクセスに利用する。上記の URL で operatorPanel は Scientific Linux 5 以前の場合で、Scientific Linux 6 以後の場合は、scripts となる。

configure method

定義

```
def configure(self):
```

説明

これはつぎのように呼び出す。

```
com.configure()
```

DAQ Operator に configure コマンドを送る。

start method

定義

```
def start(self, runNo):
```

説明

これはつぎのように呼び出す。

```
runNo = '1'
```

```
com.start(runNo)
```

DAQ Operator にラン番号を付けて start コマンドを送る。

stop method

定義

```
def stop(self):
```

説明

これはつぎのように呼び出す。

```
com.stop()
```

DAQ Operator に stop コマンドを送る。

pause method

定義

```
def pause(self):
```

説明

これはつぎのように呼び出す。

```
com. pause()
```

DAQ Operator に `pause` コマンドを送る。

resume method

定義

```
def resume(self):
```

説明

これはつぎのように呼び出す。

```
com. resume()
```

DAQ Operator に `resume` コマンドを送る。

unconfigure method

定義

```
def unconfigure(self):
```

説明

これはつぎのように呼び出す。

```
com. unconfigure()
```

DAQ Operator に `unconfigure` コマンドを送る。

getLog method

定義

```
def getLog(self, tag):
```

説明

呼び出し方には 3 通りある。1 つは、

```
all = com. getLog('all')
```

これは Log 情報すべてを `all` に入れる。もう 1 つは、Log 情報の中で指定された `tag` のみの情報を引き出す。例えば、`state` を引き出すには

```
state = com.getLog('state')
```

また、`status` を引き出すためには、
`status = com.getLog('status')`
いずれも、最初に現れた `tag` の内容を返す。

setRunNumber method

定義

```
def setRunNumber(self, runNo):
```

説明

これはつぎのように呼び出す。
`runNo = '1'`
`com.setRunNumber(runNo)`
これは `runNo` をデータベースに保存する。

cli method

定義

```
def cli(self):
```

説明

これはこのクラスをコマンドラインから実行するときに利用する。
すでに実装されている `/usr/bin/daqcom` コマンドは、下記のように実装されている。

```
#!/usr/bin/env python
from daqmwcom import daqmwcom
if __name__ == '__main__':
    daqmwcom = daqmwcom()
    daqmwcom.cli()
```

例題

下記にこのクラスを使った簡単な自動化のスクリプトを示す。

```
#!/usr/bin/env python
import sys
from daqmwcom import daqmwcom
import time
url = "http://localhost/daqmw/operatorPanel/ "
com = daqmwcom(url)
```

```

argc = len(sys.argv)
if argc != 2:
    print "usage: python test-daqmwcom.py runNum"
    sys.exit()
status = com.getLog('status')
if status != 'OK':
    print "DAQ components are not ready..."
    sys.exit()
com.configure()
time.sleep(2)
state = com.getLog('state')
if state != 'CONFIGURED':
    print "DAQ components are not configured..."
    sys.exit()
runNo = sys.argv[1]
com.start(runNo)
time.sleep(2)
state = com.getLog('state')
if state != 'RUNNING':
    print "DAQ components are not running..."
    sys.exit()
print "Now data taking start... Wait for 20 seconds and then stop..."
time.sleep(20)
com.stop()
time.sleep(4)
state = com.getLog('state')
if state != 'CONFIGURED':
    print "DAQ components are not stopped.."
    sys.exit()
com.unconfigure()
time.sleep(2)
state = com.getLog('state')
if state != 'LOADED':
    print "DAQ components are not unconfigured..."
    sys.exit()

```


参考文献

1. 千代浩司、DAQ-Middleware 1.1.1 開発マニュアル
<http://daqmw.kek.jp/docs/DAQ-Middleware-1.1.1-DevManual.pdf>
2. 仲吉一男、DAQ-Middleware 1.1.0 技術解説書
<http://daqmw.kek.jp/docs/DAQ-Middleware-1.1.0-Tech.pdf>
3. 安 芳次、Web を用いた DAQ ミドルウェア GUI 操作マニュアル
<http://daqmw.kek.jp/docs/UsersGuideOfWebGUIForDAQMW.pdf>