The *myget* User's Guide

Version 2.0

Christopher Slominski

May, 2014

Introduction

The *myget* program is used to get archived EPICS channel history. It is a UNIX shell command line program, available on *rhel-6-ia32* Linux systems. The program extracts data from a Mya channel history archive; refer to Mya documentation on the certified software web site for specific information about the Mya archiver.

The tool is most commonly used to get a sequence of channel events over a time span. It writes its results to standard output, one line per channel event. They may be channel discontinuity events or value update events. The format of an output line includes four columns; the date of the event, the time of the event, the channel value or discontinuity indication, and an optional event specific notation. The text below shows a sample of *myget* output.

```
2006-12-08 10:43:10 270.696 << Origin of channel's history >>
2006-12-08 10:43:14 270.696
2006-12-08 10:43:18 271.429
2006-12-08 10:43:22 270.94
2006-12-08 10:43:26 271.184
2006-12-08 10:43:30 269.963
2006-12-08 10:43:34 269.719
2006-12-08 10:43:38 270.94
2006-12-08 10:43:41 << Archiver shutdown >>
2006-12-09 09:00:13 253.517
```

The *myget* command should be in your UNIX environment's path variable. You may enter *myget –h* to get a brief summary of the command line options, or *myget –help* to open this User's Guide in an internet browser window. The remaining sections of this document describe these options in detail. The program uses a configuration file to modify some of the application's behavior. This file is stored in the certified software repository and is not available for change by regular users of the *myget* software.

Designating the Time of Interest

You may need to specify times of interest when making various channel history requests. The absolute time format is "[YYYY-MM-DD][HH:MM[:SS]]"; Optional fields are designated by *[]* pairs. If a date is not supplied, today is assumed. If a time is not supplied, 00:00 is assumed. Remember to place the full absolute time string in quotes on the UNIX command line since it contains a space character between the date and time; and would be treated as two separate arguments otherwise.

Relative time may also be specified, and is interpreted as a signed offset from now. You may append a units specification character to the numeric offset {s:second m:minute h:hour d:day w:week}, but if no units are specified on the relative offset, seconds are assumed. Note a positive offset means some time in the future and a negative offset is in the past. For example, *"-12d"* represents the date twelve days before now.

When specifying a time range on the command line, you designate the beginning and ending times of the span. You may enter these times as described above, however you may also want to enter a time offset that is relative to the beginning (or ending) time instead of being relative to now. This is done by adding a '^' prefix to your integer offset, which is then interpreted as relative to the other side of the time span. For example, the command line options "*–b 12:00:00 –e ^1h*" specifies a one hour time span beginning at noon today; using the command line begin (-*b*) and end (-*e*) options.

Choosing a Mya system

There may be more than one Mya system installed within the Jefferson Lab computing environment, and you may use *myget* to fetch channel history from any one of them. The program will default to the Mya system selected in the program's configuration file, which is typically the operations department Mya deployment. You will seldom need to choose a particular Mya installation, however to do so you use command line options. You can specify a known Mya deployment with the *–m* option. Use *myget –h* to see which Mya deployment is the default. The *myget* configuration file contains a path to a directory containing a set of files that describe individual Mya deployments. When you use the **–m** option, you are actually selecting one of the files in this directory.

You may also specify a Mya system by the host name of the Mya Master, using the *–M* option. This is not a standard technique and is available for software developers who may have a test Mya system installed on some development machine. The format of the option is *"–M* <host name>".

EPICS Enumerations

EPICS channels that are defined as an enumerated data type (*DBR_ENUM*) have their values stored by Mya as the integer enumeration index. The integer values are shown when requests are made for either a single channel update event or the events occurring over a time span. Users may wish to see the enumeration text associated with the integer index. To do so, use the **-s** option on the command line, which cause the lookup of the EPICS enumeration definition.

Requesting Channel Values over a Time Span

```
Get all channel history occurring during a span of time.

Usage: -c<opt> [-b<opt>] [-e<opt>] [-l<opt>] [-p] [-m<opt>] [-M<opt>] [-d[<opt>]] [-
f<opt>] [-s]
c; Channel name
b; Begin time; default = -60
e; End time; default = 0
l; Limit number of events returned
p; Include prior point
m; MYA deployment tag; default = ops
M; MYA master host; overrides -m
d; Data events only; optional filter expression
f; Fractional time digits; default = 0
s; Enumerations as strings

Examples:
  myget -c CHAN1 -b -2d -e0          # Get most recent 2 days of history
  myget -c CHAN1 -p                  # Get default span plus prior event
  myget -c CHAN1 -b12:00:00 -e^+1m   # One minute span starting at noon
  myget -c CHAN1 -b '2008-01-01 12:00:00' # Absolute begin date
  myget -c CHAN1 -d 'c > 2'          # Data values greater than two.
```

Asking for all of the channel update events that occurred during some time span is likely the most common usage of *myget*. The only required argument is the channel name. The beginning and ending times have defaults, which may be viewed when requesting command line help. The format of time entry is described in a previous section, as is an explanation of how to specify a Mya deployment other than the default. The text box above shows the command line help output.

There may be a huge number of channel events contained within a very long time span, such as a year. If the time required to fetch and process the events is prohibitive, you can request a reduced data set with the *–l<n>* option. In this case, *myget* partitions the time span into 'n' equally size time segments, extracting just an event from each. Note that some segments may not have an event, so the actual number of events returned may be less than the number you specified with the option.

There may be no events occurring within a requested time span, which leads to a warning diagnostic from the *myget* program. This occurs when significant channel changes occur infrequently relative to the length of the specified time span. Use the *–p* option to include the most recent channel event that occurred prior to your time span window to guarantee you will at least get one event. If there are no other events except the prior point, then the value of the prior point persisted throughout your time span of interest.

The default output time format has a resolution of seconds. If you are interested in a finer resolution, you may use the *–f* option to request a number of fractional digits to be displayed.

Data only and filtering by logical expression

Use the –*d* option when you do not want disconnection events included in your output. This will get a series of the values attained with no indication of when the value of a channel was not known.

You may also supply a logical expression with the –*d* option. You do this when you only want to show data events that match some criteria. For example "-d '$abs(c) > 2'" limits the output channel history values to only those that are greater than 2 in magnitude. You use the character 'c' to refer to the channel name that you supplied with the –*c* option.

The expressions you provide are implemented using a generic expression evaluation library facility. You can read about the details of valid expressions at this [web site].

Requesting Channel Values at a Point in Time

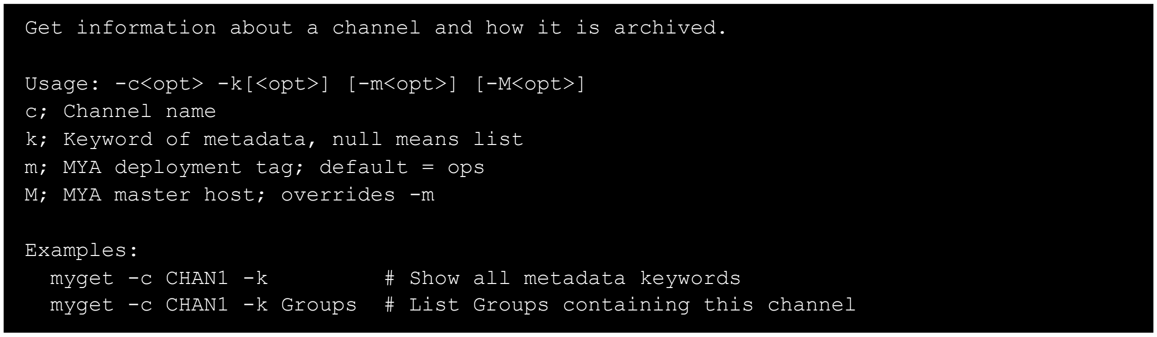
```
Get a single history event near a specific time. You specify which way
to look; inclusive before (-) or exclusive after (+).

Usage: -c<opt> -t<opt> [-w<opt>] [-m<opt>] [-M<opt>] [-d] [-f<opt>] [-s]
c; Channel name
t; Time of interest
w; Which way: use + or -; default = -
m; MYA deployment tag; default = ops
M; MYA master host; overrides -m
d; Data events only
f; Fractional time digits; default = 0
s; Enumerations as strings

Examples:
  myget -c CHAN1 -t "2007-11-29 14:00:00" # Event at or before date
  myget -c CHAN1 -t0 -w-                  # Get last recorded event
  myget -c CHAN1 -t 1970-01-01 -w+        # Get first recorded event
```

This type of request is similar to a time span request, but you supply only one time of interest and *myget* returns the closest event either "*at or before*" or "*after*" the specified time. You use the –*w* option to indicate before (-) or after (+). Other command line options are the same as described above. You get one line of output from a successful execution of this command. It will either be a description of the identified event or a diagnostic indicating that there is no event available. The command line help for this type of request is shown in the text box above.

Requesting Channel Metadata



```
Get information about a channel and how it is archived.

Usage: -c<opt> -k[<opt>] [-m<opt>] [-M<opt>]
c; Channel name
k; Keyword of metadata, null means list
m; MYA deployment tag; default = ops
M; MYA master host; overrides -m

Examples:
  myget -c CHAN1 -k          # Show all metadata keywords
  myget -c CHAN1 -k Groups  # List Groups containing this channel
```

An archived EPICS channel may have metadata associated with it. A common example of this is the set of text strings associated with an EPICS enumerated data type. The history for these channels appears as integer values in the normal *myget* output. You may wish to know the text strings associated with the enumerations, and can get this information with the **–k** option. Metadata is categorized by keywords. For example, the keyword *enum_strings* designates the set of enumeration text strings of a channel. To get metadata for a channel, you execute *myget* with the **–c** and **–k** options set to the channel and keyword names respectively. You do not specify times of interest with metadata requests. If you do not know the metadata keywords defined for a channel, just issue the command with no qualifier after the **–k** option. The software will list all of the keywords for that channel.

Note that metadata may be shown with or without a timestamp, depending on the type of metadata requested. Some of the Mya metadata is configuration information used by Mya to archive a channel. This information does not have a time history and only the current value is available. Other metadata can be found as a time sequence of settings and will be shown as such.

Also note that some metadata keywords will have a set of data associated with them. This is the case with the enumerations strings. Each of the strings is output when the keyword is requested.

Requesting Channel Correlations

```
Correlate a set of dependent channels against one independent channel.
You may explicitly choose the correlation style using the -s option
as Interpolate, Prior, Next, or Closest.

Usage: -c<opt> [-b<opt>] [-e<opt>] [-l<opt>] [-m<opt>] [-M<opt>] [-s<opt>] <Channel>
...
<Channel>; Dependent channel to correlate
c; Independent channel
b; Begin time; default = -60
e; End time; default = 0
l; Limit number of events returned
m; MYA deployment tag; default = ops
M; MYA master host; overrides -m
s; Correlation style; default = Prior

Examples:
  myget -c R2XXITOT IBC0R07CRCUR1 # Correlate over default time span
  myget -c R2XXITOT -l1000 abc    # Limit to a maximum of 1000 points
```

You can get correlation information from *myget* by supplying the name of an independent channel along with the names of the dependent channels. The dependent channels appear as unqualified command line arguments. With the exception of the –*s* option, the command line options follow the explanations found previously in this document.

Note that output will be ordered by ascending value of the independent channel. All values in a row are separated by a single white space. This is fine, and efficient, for the normal use case where you are redirecting the typical reams of output into a file for import into a spread sheet. If you want to make your output pretty for screen viewing use some standard output UNIX filer such as *column* (linux), as shown below. The output shown below has 'N/A' in some cells of a dependent channel. These mean the dependent channel had no value for the given independent channel value.

The user may specify a particular correlation style on the command line. This tells *myget* how to determine a representative value of the dependent channel when a particular dependent event occurs. It is highly unlikely that the dependent channel gets an update at the exact same time as the dependent channel, therefore a representative value of the dependent channel must be selected. The choices will use the dependent events that occurred before and/or after the time of the independent event. The style indicates how to use the surrounding events to select a representative value. The default is to use the prior event, but you can request to use the next, closest, or a linear interpolation of the two.

```
> myget -cMMSENTRY -b -10w IPM0I07.XPOS VIP0F03log | column –t

MMSENTRY  IPM0I07.XPOS  VIP0F03log
400       N/A           1.99562e-10
400       N/A           2.0699e-10
400       N/A           2.04771e-10
401.9     -2.43644      1.65344e-10
402       0             1.63994e-10
414.5     N/A           6.23156e-10
442.5     N/A           2.46571e-10
500.08    N/A           2.01535e-10
```

```
500.08    N/A           2.05877e-10
500.08    N/A           1.96087e-10
500.08    N/A           2.09281e-10
500.08    N/A           1.89006e-10
510       -2.4877       2.08498e-10
510       N/A           1.7299e-10
550       -2.43644      1.66471e-10
589       -2.43644      1.74251e-10
```

Program Exit Status

The *myget* program returns exit status to the command shell. There are three possible values; 0, 1, and 255 that correspond to normal, warning, and error status respectively. Normal status means you entered a valid request and have received data. A warning is issued for a valid request that output likely unexpected results, such as no values available in the time specified. The error status is returned for serious errors like using an undefined command line option, the Mya server being down, or requesting a non existent channel.