Program *Archive*

User's Guide

Version 4.2 – Christopher Slominski

November 2012

Introduction

This *Archive* program provides a means to view the channels being archived by the Mya channel archiving system, and to request additional EPICS channels be archived by Mya. The general user makes requests to add to the *User Channel Set*, as opposed to the *Core Channel Set*, which is managed solely by the Mya administrator. Review Mya documentation to familiarize yourself with concepts about the two distinct categories of archived channels. This document was written primarily for the general user and therefore focuses on managing the user channel set. Since the same tool is used to manage the core set, an appendix is included on the topic of adding channels to the core set, as a reference guide for Mya administrators.

The program is a command line utility, available on Linux systems that have network access to the Mya deployment you intend to address. There are no quotas imposed by the tool, but please be judicious in making your requests. We would like to keep the total user set to less than 10,000 channels, so your asking for a thousand would command an unfair share of the resource.

Note that for a control system channel to be archived, it must be visible on the sub-net to which Mya listens. The OPS deployment of Mya cannot see all control system fiefdoms in use at Jefferson Lab. A good way to check if the channels you want to archive are suitable is to log into a machine on the OPS fiefdom and perform a *caget* command. There is an independent DEV deployment of Mya that has connectivity to selected IOCs on the DEV network.

User Set Requests

Remember that channels archived via the user set are of limited duration. Only the most recent span of channel history is kept by Mya, and the archiving of a channel expires after a certain amount of time. The default attributes and their upper limits are configurable with the installation of the *Archive* program and will be shown when you perform an "*archive –h"* command.

Requests made using the tool do not go into effect immediately. The Mya system does not put a large emphasis of servicing new archive requests. Mya pauses from some of its activities every 30 seconds to perform a number of small administrative chores. During this time, Mya will service requests to archive new channels, but limited to 100 requests. The next 100 requests will have to wait until the next administrative pause. Note the 30 second and 100 request attributes of Mya are configurable with a Mya deployment's installation. The values cited here reflect the values associated with the OPS deployment as of the writing of this document.

Using the tool

There are four distinct usages of the utility, which are identified by the options and arguments provided by the user on the command line. These are referred to as command line signatures. Enter "*archive -h*' to get the syntax of the various signatures. Note you may also request this

manual be shown in your browser with the *–help* flag. One command line signature pertains to core set updates, which may only be requested by the Mya administrator, as the usage is password protected. The syntax for these types of requests is discussed in the document's appendix.

Note that all signatures, except the help request, take one or more unqualified command line arguments. These arguments are either group names, channel names or the names of files that contain channel names. File names are denoted by a leading '@' character prefix. The format of a channel specification file is one channel per line. Optional information may be appended on the line after the channel name depending on usage, but must be separated from the channel name by white space.

The sections below provide their command line signature syntax. Note that anything in square brackets is optional and an ellipsis (…) means one or more.

Channel lookup

One command line signature is used to lookup archive status of channels. You simply list the names of channels or files on the command line. If by file, any information on the lines of the file after the channel name is ignored. One optional command line switch for this signature, the –m option, is used to identify a Mya deployment other than the default. Users will seldom need to specify a non default Mya deployment. The default deployment is shown in the command line help information.

```
Usage: [-m<opt>] <spec> ...
<spec>; Channel or file name
m; Mya deployment title; default = ops

Examples:
  archive R247GSET  # Is channel archived?
  archive @file     # Are channels in file archived?
```

Archive Group lookup

You may look up archive group information using another command line signature. You get a list of all groups when no qualifier to the option is given. If the user supplies the name of an archive group, information about all of the channels in that group is given.

```
Usage: -g[<opt>] [-m<opt>]
g; Group name; null means list
m; Mya deployment title; default = ops

Examples:
  archive -g        # List all archive groups
  archive -g bpm    # List channels in the bpm group
```

Adding to the User Channel Set

Another command line signature is for requesting the addition of channels to the user channel set. It is differentiated form other signatures by the required –u flag. You may specify a Mya deployment other than the default with the –m option, but likely will never need to do so.

You may specify an archiving dead band for the channels you are requesting. This limits archiving of channel changes to only those above the specified dead band. Note the dead band is applied to all channels given on the command line, as well as channels specified in files that do not have an explicitly stated dead band following the channel name. See Mya documentation for details of various formats for dead band requests.

As previously stated, user set requests are of limited duration. Mya will only keep the most recent history of a channel and will discontinue archiving the channel once the expiration period is reached. You may specify the keep span (-k) and expiration time (-e) on the command line, or take the defaults that the program provides. You may not request periods longer than the installation's imposed limits. You can see what the defaults and limits are set to by requesting command line quick help (*-h*).

The format of a channel request file allows for an optional dead band to be specified after each channel name, separated by white space. Anything other than a valid dead band on the line will be considered a fatal error.

Note that it is not an error to request archiving of a channel that is already being archived. If it is a member of the core set, your request just has no effect on the archiver. You've asked for a channel to be archived, and indeed it is. If the existing channel is in the user set, you've staked another claim against it and the archiving of the channel only expires when all requests for that channel have expired, including yours. There is a subtle difference though. Archiving attributes are defined by the first request for a channel. Additional request for the channel may request dead bands and keep spans that are silently ignored. It's best to always check to see if, and how, a channel is being archived before making a request.

```
Usage: -u [-m<opt>] [-d<opt>] [-k<opt>] [-e<opt>] [-E<opt>] <spec> ...
<spec>; Channel or file name
u; User channel set update
m; Mya deployment title; default = ops
d; Deadband
k; keep span (days); default = 7
e; Expiration time (weeks); default = 15
E; EPICS_CA_ADDR_LIST; default = 129.57.255.21

Examples:
  archive -u R247GSET  # Start archiving one channel
  archive -u @file     # Start archiving a set of channels
```

Error handling

The utility takes the "all or nothing" approach to archival requests. If any errors are detected while processing your request, the entire request is rejected. This makes repeating large requests easier, after correcting some problem, as you do not have to figure out which parts of the request were granted and which were not. Just resubmit the request.

Part of validating a request includes connecting to all requested channels in the control system. Mya needs some channel metadata for those channels requested and gets it directly from the control system. This also serves as a good means for keeping the archiver uncluttered by

attempts to archive misspelled channels. The request is rejected if all channels do not connect. The unfortunate side effect to this is that users will not be able to proactively request channels that they know will be available in some soon to be installed EPICS application.

Expiration notification

The user requests you make will eventually expire. When they do, you will receive an email notification of the occurrence. The email is sent to 'login'@jlab.org, where 'login' is the user identification from the workstation you were logged into when you made your very first ever user set request. That's right, an account is made for you automatically the first time you use the tool to make a user request. Your login is saved as well as the standard Jefferson Lab email address. The ramification of this is, when making archive requests from someone else's login, they will be the owner of the request and will get expiration notifications.

If you do not want expiration notifications, contact the Mya administrator to disable notifications for your account.

Appendix A - Core set requests (Mya administrator only)

The *archive* utility may be used to add channels to the Mya core set, but only by an administrator armed with the required password. You specify channels the same as you do in the signatures described earlier in this document, but the command line signature differs when making a core update request. It is a good idea to submit the channels to a *lookup* to see which ones may already be archived. You may need to work out discrepancies depending on what is intended.

You identify a core set update by use of the –c (core) and –g (group) flags. Additionally you may provide the -i flag, which will identify the Mya instance that will host the channels; obtaining and storing the channels control system values. This does not have to be the Mya Master instance. The master instance database does get updated because it contains information about all channels being archived. It gets the master instance specification from the chosen Mya deployment. You specify an instance by host name, which must not include domain information, so "opsmya1" is used instead of "opsmya1.acc.jlab.org".

The group flag indicates which archive group you want the channel to be referenced by. You can create a new group by giving a name that does not already exist; when doing so you will be prompted to confirm you want to create a new archive group. You may request channels that already exist, though this is best restricted to those that are only in the core set. Your new group will just share references to the channels being archived. This is a good way to make small specialty group. You may have a large group having references to all BPM channels. You could then make a NL_BPM group that references just North LINAC BPMs. When you use the utility in this way, there is no need to specify a Mya instance.

You may also provide the Mya deployment and dead band as described above under user set requests. There are two other options not available with the user set signature. The EPICS native

data type (-t) and array element count (-s) can be include on the command line. As with the user set, the tool will still connect to EPICS to look up a channel's data type and array size information. When it can't connect channels, it will use the command line values if provided. This allows for the installation of channels that will exist in the future, trusting a disciplined administrator not to mistake misspelling with future channels.

A Mya deployment have been installed to have an additional channel table partition besides the default used my MySQL. Users of the *archive* tool may designate that the new core channels be added to a specific disk partition by providing the partition ID, using the –p option. If not supplied, the application will choose the least loaded disk partition. You may see the partition IDs on a running Mya instance via the *bin/mahout disk* command. Otherwise you must look at the *partitions* table in the database.

Although not normal for the core channel set, you are allowed to provide a keep span and group expiration time just like user set requests. The default for these differs for the core set however, having no keep span or expiration time by default.

Auto Load Balancing

If a target instance is not provided on the command line, the utility will choose one for you. It does so be examining each Mya instance in a deployment, excluding the *sandbox* instance, to determine which has the least loaded Mya disk partition. All channels in the request are added to the identified instance; there is no load balancing per channel in one request. Automatic Load balancing occurs for redundant archiving as well.

An administrator typically selects a particular instance when adding channels to an existing Archive Group; keeping all of a group's channels on the same instance. The tool should be allowed to choose when creating a new Archive Group.

Redundant Archiving

Mya deployments may be associated with a redundant Mya deployment through the deployment descriptor files. When this is the case, the *archive* tool will automatically repeat all core set requests on the redundant system. It will appear to the user as two sets of output from the program, confirming the actions taken on both Mya deployments.

When the redundant deployment has more than one instance, load balancing as described above occurs.

EPICS environment

When requesting the archiving of channels, the application looks up characteristics of the channels in the EPICS control system. It is important that the channels are looked up in the proper control system context. The context is set in a process' environment by setting the variables EPICS_CA_ADDR_LIST and EPICS_CA_AUTO_ADDR_LIST (always equal 'NO').

The application does not assume the user's environment is set up correctly for the request being made. It sets the environment variables appropriately for the chosen Mya deployment, however

users have control over this with the '-E' command line option. The best strategy is to extract from the Mya master host's configuration file the environment settings. This is the default for core set requests. Since the environment extraction is a privileged operation, general users cannot use this strategy. Therefore the user set request defaults to the IP address 129.57.255.21, which is the OPS gateway and what is currently configured for the OPS Mya master.