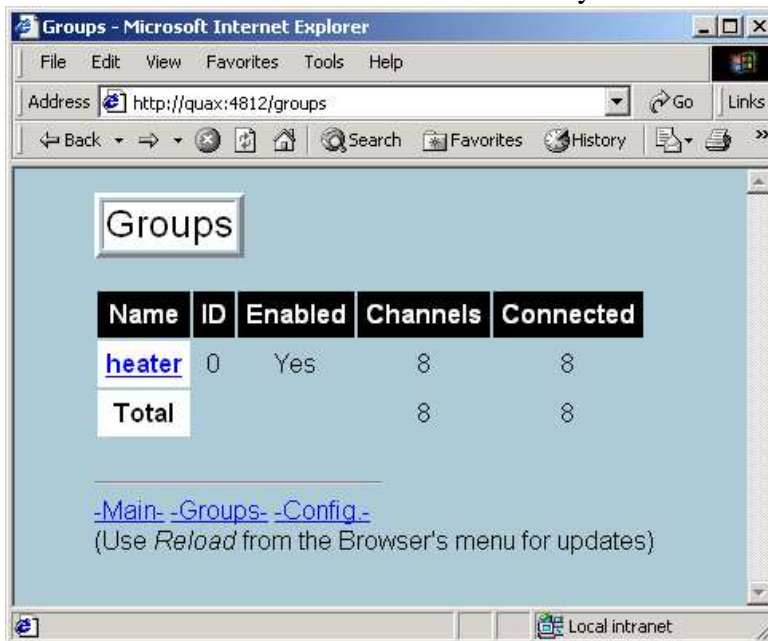


Archive Exercises for Water Tank and Heater

Exercise 1: ArchiveEngine, simple retrieval

Generate an archive configuration file for your water heater simulation: Setpoint, readback (tank temperature), heater voltage, ...

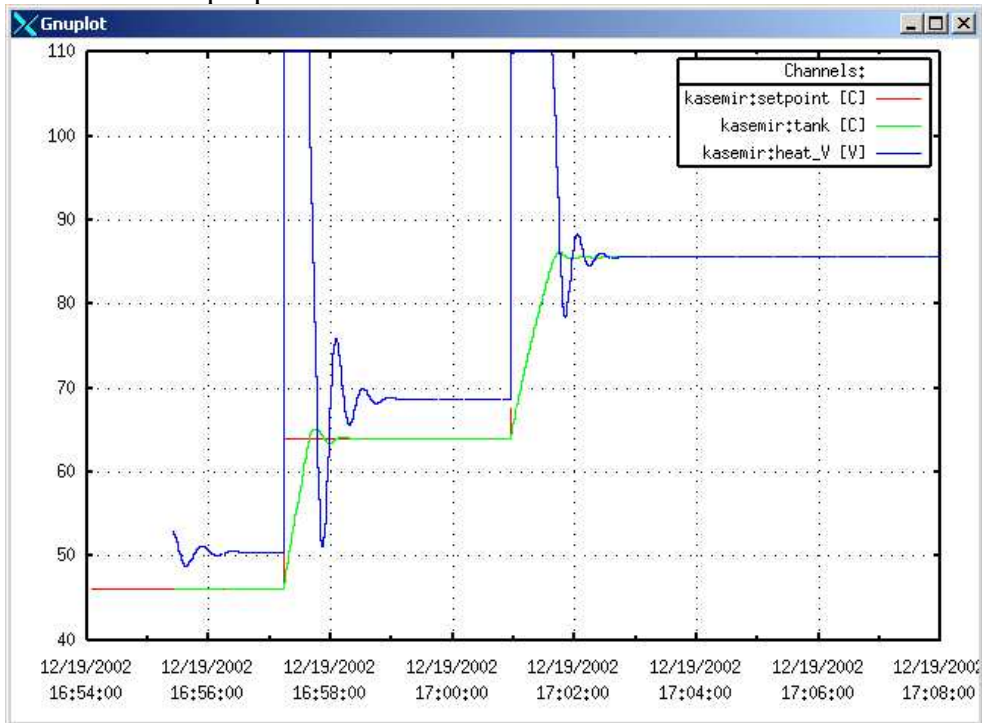
- Run the ArchiveEngine with this configuration.
- Check the online-status. You should find all your channels connected:



- Run ArchiveManager:
Check what time range that your archive covers.
Dump the raw values for the readback channel.
- Run ArchiveExport:
Retrieve setpoint, readback and heater voltage in “gnuplot” format.
You should find two files: A raw data file and a gnuplot command file, example:

```
>ls plot*
plot plot.plt
>gnuplot
      G N U P L O T
      Version 3.7 patchlevel 2
...
gnuplot> load 'plot.plt'
```

This is an example plot:



- Run ArchiveExport:
Generate a spreadsheet for setpoint and readback with and without the '-fill' option.

Exercise 2: CGIExport

Run a web browser, go to the URL for the CGIExport demo page, and enter the path to your archive.

You should be able to do the same basic queries as in the previous example. Most people prefer this "user interface", accessible from every web browser, and accept the reduced speed.

- Check what time range your archive covers.
- Retrieve values for the readback channel.
- Generate an online-plot of setpoint, readback and heater voltage
- Download a spreadsheet-file of setpoint, readback and heater voltage, with and without the 'fill' option.

Channel Archive - Microsoft Internet Explorer

Address: http://quaxy/archive/Default.htm

Channel Archive CGI Interface

The following examples show some ways of linking to the CGIExport tool from a web page. You should try to understand the HTML text and then come up with your own web pages. The links on this page assume that the web server was setup according to the ChannelArchiver/doc/cgiexport/setup.html.

CGIExport: Supported CGI parameters

To see if the web server is configured correctly to execute the CGIExport tool at all try a link like this which will cause the tool to display supported parameters:

<cgi/CGIExport.cgi?COMMAND=HELP>

Take 1

This example allows you to start the CGI interface for a user-supplied archive file:

Archive:

EPICS Channel Archive - Microsoft Internet Explorer

Address: ENDMINUTE=42&ENDSECOND=59&COMMAND=GET&FORMAT=PLOT&Y0=0&Y1=100&REDUCE=ON&INTERPOL=0

Channel Plot

Channel List

- kasemir:PID (I)
- kasemir:setpoint (C)
- kasemir:tank (C)

Pattern:

file glob

Names:

Start: Day (m/d/y) 12 19 2002 Time (h:m:s) 18 36 00

End: Day (m/d/y) 12 19 2002 Time (h:m:s) 18 42 59

Plot, v limits 0 100 All Data: (default: reduced to plot size)

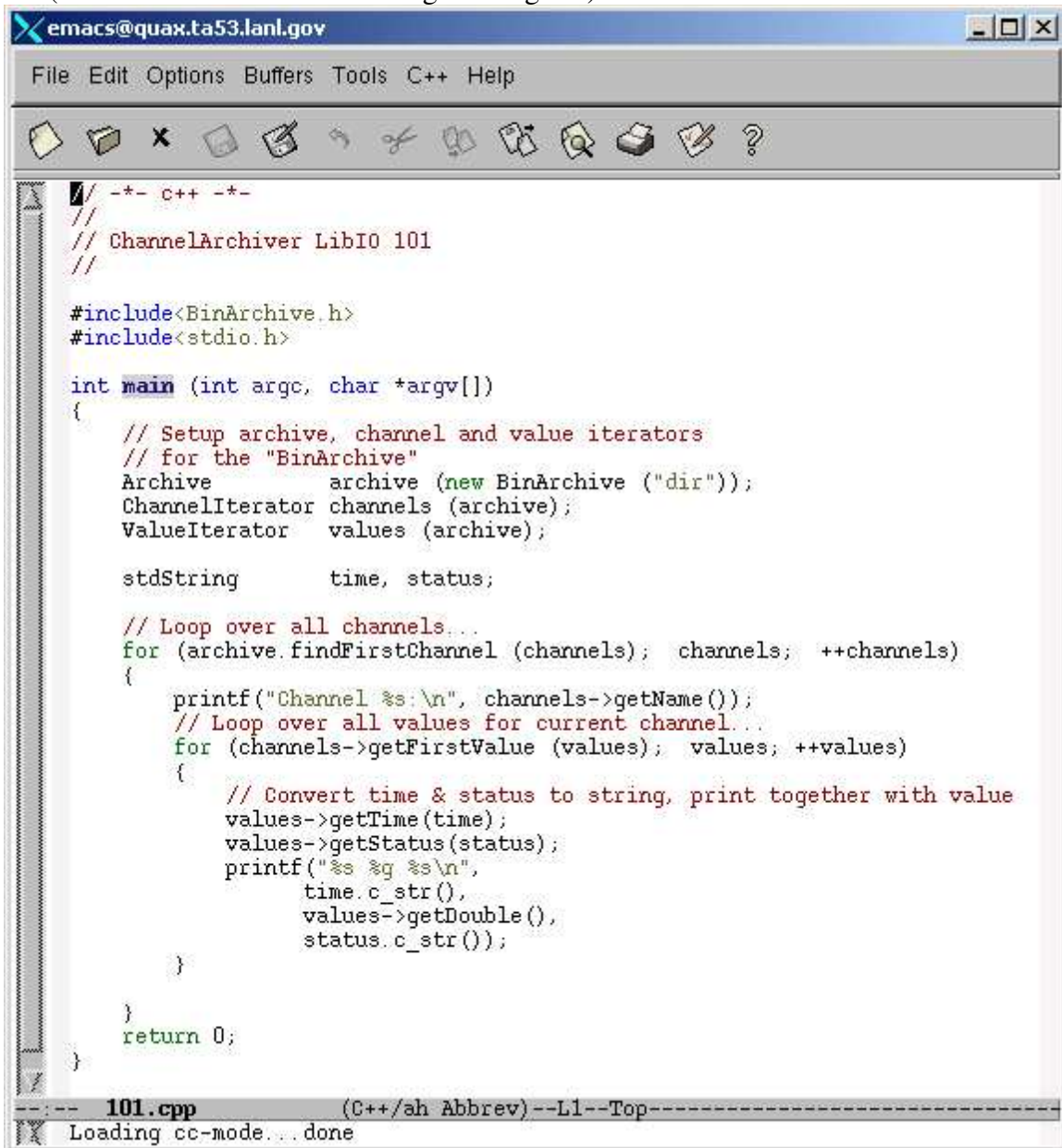
Exercise 3: More Data Analysis

Compile and run the "101.cpp" C++ program which dumps all values for all channels. Change it to do the following:

- Iterate only over your "temperature readback" channel, calculating the average, minimum and maximum temperature

For the following, you have to handle two process variables and deal with the time stamps. Decide if this is easiest done by modifying the C++ code or by first dumping the data with ArchiveExport and then analyzing it with a separate program (which in turn can be C++ or perl or ...)

- Determine for how long the tank temperature was below, above and at the setpoint (within a certain deadband of e.g. 0.5 degrees)



```
emacs@quax.ta53.lanl.gov
File Edit Options Buffers Tools C++ Help

// -- c++ --
// ChannelArchiver LibIO 101
//

#include<BinArchive.h>
#include<stdio.h>

int main (int argc, char *argv[])
{
    // Setup archive, channel and value iterators
    // for the "BinArchive"
    Archive      archive (new BinArchive ("dir"));
    ChannelIterator channels (archive);
    ValueIterator values (archive);

    stdString    time, status;

    // Loop over all channels...
    for (archive.findFirstChannel (channels); channels; ++channels)
    {
        printf("Channel %s:\n", channels->getName());
        // Loop over all values for current channel...
        for (channels->getFirstValue (values); values; ++values)
        {
            // Convert time & status to string, print together with value
            values->getTime(time);
            values->getStatus(status);
            printf("%s %g %s\n",
                time.c_str(),
                values->getDouble(),
                status.c_str());
        }
    }
    return 0;
}

-- 101.cpp (C++/ah Abbrev)--L1--Top-----
Loading cc-mode... done
```

```
emacs@bogart.ta53.lanl.gov
Buffers Files Tools Edit Search Mule Help
# Skeleton code for running 'ArchiveExport'
# and parsing the result, one line at a time,
# in a perl script

open DATA, "ArchiveExport -fill dir kasemir:setpoint kasemir:tank |"
  or die "Cannot open data source\n";

while (<DATA>)
{
  # Skip anything that doesn't start w/ time stamp
  next unless m/^[0-9].*/;
  # Parse date, time and two channels (in this case, setpoint and readback)
  ($date,$time,$setpoint,$readback) = split;
  # Decode the time stamp (Note: ignoring the date!)
  if ($time =~ m/^[0-9][0-9]:([0-9][0-9]):([0-9][0-9]\.[0-9]*)/)
  {
    $h = $1;
    $m = $2;
    $s = $3;
    $seconds=($h*60.0+$m)*60.0+$s;
  }
  else
  {
    die "Cannot decode time string\n";
  }
  print "$seconds - $time - $setpoint - $readback\n";
}

--:-- 101.pl (Perl Abbrev)--L27--A11-----
```