

# Classic Channel Archiver 설정 메뉴얼

손 창욱\*

Rare Isotope Science Project  
Institute for Basic Science, Daejeon, South Korea

March 23, 2015

## Abstract

Classic Channel Archiver는 EPICS의 ChannelAccess를 통해 각 device에 대한 Channel정보를 일련의 파일로 저장하는 툴이다. 본 메뉴얼은 Classic Channel Archiver의 Main Engine인 ArchiveEngine 및 Engine Configuration, 저장된 Data를 Retrieval하기 위한 DataServer Configuration을 포함하고 있으며 Channel Archiver Manual[1]을 바탕으로 작성되었다.

## 1 Sample Mechanisms

### 1.1 Monitored Sampling

Monitor Mode의 경우 Archive Engine(AE)은 Channel Access(CA) 서버에게 monitor요청을 하게 된다. 즉, CA 서버는 채널 값이 변경되면 AE에게 알리게 되고 이 때 채널 값을 저장하게 된다. 결과적으로 Monitor Mode는 채널 값이 변경될때 마다 저장하는 방식이다. 이 모드의 경우 다음 몇 가지 문제점이 있다.

- 채널 값이 변하지 않는 경우 Archive가 되지 않는다.
- 채널 값이 빠르게 변할 경우 데이터 용량이 커진다.

---

\*scwook@ibs.re.kr

## 1.2 Scanned Sampling

Scan Mode의 AE는 채널 값을 CA 서버에게 주기적으로 요청한다. 이 경우 Monitor Mode와 달리 채널 값이 변하지 않더라도 값을 저장하게 된다. 하지만 동일한 값이 계속 저장되면 저장공간 낭비가 발생 하므로 AE는 이전 값과 같은 값이 반복되어 들어오면 'repeat count'를 저장하여 공간을 절약한다. Scan Mode의 경우 다음과 같은 문제점이 있다.

- 'fault' 정보를 가지고 있는 채널에 대해서는 적합하지 않다. 만약 'fault'가 scan 주기 사이에 발생하게 되면 AE는 이 값을 저장할 수 없다.
- AE는 CA 서버에게 주기적으로 요청하므로 네트워크 부하가 발생할 수 있다. 따라서 Scan Mode는 1분 또는 그 이상의 주기를 가지는 Archive에 대해 적합하다.
- 네트워크 지연 및 record의 주기등과 같은 이유로 채널의 시간에 오차가 발생할 수 있다.

## 1.3 Scanned using Monitors

Scan과 Monitor Mode를 합쳐놓은 것으로 AE의 설정에 따라 Scan 또는 Monitor Mode로 작동한다. 예를 들어 IOC의 record값이 1초 주기로 변경되는 경우에 대해 30초 마다 저장하고자 하면 Scan Mode가 효율적이다. 하지만 만약 5초 마다 저장할 경우 네트워크 부하를 줄이기 위해 Monitor Mode가 더 적합하다. 즉 CA 서버로 부터 1초마다 채널 정보를 받아 1개의 값만 저장하고 4개의 값을 무시하는 방법이 효율적이다. 이 방법은 Scan Mode에서 작동하며 AE의 설정 값중 get\_threshold 값에 따라 결정 된다. 만약 Scan 주기가 get\_threshold 보다 작을 경우 Channel 값은 Monitor Mode로 전환되어 작동한다. 이 방법은 다음과 같은 문제점이 있다.

- 1 또는 30초 주기를 가지는 Scan의 경우 일반적인 Scan Mode보다 네트워크 부하가 작은 장점이 있지만 여전히 'repeat count'에 대한 정보를 위해 CA 서버에 채널값에 대한 요청이 필요하다. 만약 0.1초 주기와 같이 빠른 경우 일반적인 Monitor Mode에 비해 CPU 부하가 커질 수 있다.

## 2 설치

Channel Archiver를 설치하기 위해서는 기본적으로 EPICS Base 및 Extension이 설치되어 있어야 한다. 본 메뉴얼에서는 다음 EPICS 구조를 따라 설치를 진행하였다.

- EPICS Structure

```
ctrluser@ctrluser $  
epics  
├── downloads  
└── R3.14.12.4  
    ├── base  
    ├── epicsLibs  
    ├── extensions  
    └── siteApps
```

Channel Archiver의 기본 설치하는 홈페이지[2] 또는 Web Repositories[3]에서 소스코드를 다운 받은 후 컴파일 하면된다. 현재 CA는 2006년 이후 새로운 버전이 개발되지 않고 있어 Update된 Library 및 Compiler환경에 따른 Compile 오류가 많이 발생한다. 따라서 현재 개발 환경에 맞는 코드 수정이 필요하며 다음 Script를 이용하면 쉽게 설치가 가능하다. Script실행에 앞서 필요한 Library를 설치한다.

```
ctrluser@ctrluser:~$ aptitude install libxmlrpc-c++4-dev libxerces-c2-dev libcurl3-dev
```

script를 실행하여 설치를 진행한다.

```
ctrluser@ctrluser:~$ bash chanarch.sh
```

설치가 완료되면 extension/src/ChannelArchiver에 소스코드가 설치되고 extension/bin/linux-x86\_64 안에 Archive Engine를 포함한 실행파일들이 위치하게 된다.

## 3 Archive Engine

### 3.1 Configuration

Archive Engine을 시작하기 전 저장하고자 하는 Channel리스트, 주기, 데이터 파일 크기와 같은 기본 설정이 필요하며 이러한 설정들은 XML-type의 파일에 정의된다. XML파일을 만들기 앞서 파일 구조를 설정하는 engineconfig.dtd파일을 만든다.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- DTD for the ArchiveEngine Configuration -->
<!-- Note that we do not allow empty configurations: -->
<!-- Each config. must contain at least one group, -->
<!-- and each group must contain at least 1 channel. -->
<!ELEMENT engineconfig ((write_period|get_threshold|
                        file_size|ignored_future|
                        buffer_reserve|
                        max_repeat_count|disconnect)*,
                        group+)>
<!ELEMENT group (name,channel+)>
<!ELEMENT channel (name,period,(scan|monitor),disable?)>
<!ELEMENT write_period (#PCDATA)><!-- int seconds -->
<!ELEMENT get_threshold (#PCDATA)><!-- int seconds -->
<!ELEMENT file_size (#PCDATA)><!-- MB -->
<!ELEMENT ignored_future (#PCDATA)><!-- double hours -->
<!ELEMENT buffer_reserve (#PCDATA)><!-- int times -->
<!ELEMENT max_repeat_count (#PCDATA)><!-- int times -->
<!ELEMENT disconnect EMPTY>
<!ELEMENT name (#PCDATA)>
<!ELEMENT period (#PCDATA)><!-- double seconds -->
<!ELEMENT scan EMPTY>
<!ELEMENT monitor EMPTY>
<!ELEMENT disable EMPTY>
```

이제 앞서만든 DTD파일 구조를 따르는 engineconfig.xml파일을 생성한 후 값을 설정한다.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE engineconfig SYSTEM "engineconfig.dtd">
<engineconfig>
```

```

<write_period>30 sec</write_period>
<get_threshold>10 sec</get_threshold>
<file_size>30</file_size>
<ignored_future>1 hour</ignored_future>
<buffer_reserve>3</buffer_reserve>
<max_repeat_count>120</max_repeat_count>
<group>
  <name>AB</name>
  <channel><name>ab:TEM</name>
    <period>1 sec</period><scan/>
  </channel>
  <channel><name>ab:HUM</name>
    <period>1 sec</period><scan/>
  </channel>
</group>
</engineconfig>

```

## write\_period

그룹 및 채널값을 데이터 파일에 쓰는 주기를 지정한다. Archive Engine은 일정시간 Buffer에 값을 저장하는데 여기에 설정된 시간이 지나면 데이터 파일에 값을 저장한다. 예를 들어 30 sec로 설정 할 경우 30초 동안 들어오는 값들을 메모리 버퍼에 저장했다가 매 30초 마다 버퍼에 있는 값들을 파일로 저장한다.

## get\_threshold

scan mode일 경우 monitor mode로 변경되는 기준을 get\_threshold 따라 결정한다. 변경되는 기준은 period가 get\_threshold 보다 작을경우 monitor mode로 변경된다. 예를 들어 get\_threshold가 10sec로 설정되어 있을 때 period가 5 sec인 scan mode의 channel은 monitor mode로 작동한다. 반면 period를 30 sec로 변경 할 경우 scan mode 그대로 작동한다.

## file\_size

데이터 파일 크기를 지정한다. 예를 들어 100으로 설정 할 경우 데이터 파일이 100MB가 되면 새로운 파일을 생성 한다.

## ignored\_future

Channel로 부터 저장되는 시간이 “현재 시간 + ignored\_future” 시간 범위를 벗어나는 경우 들어오는 Channel값들을 무시한다.

## buffer\_reserve

메모리 버퍼 크기를 지정한다. 전체 버퍼 크기는 다음 수식에 따라 정해진다.

buffer reserve x write period / scan period

## max\_repeat\_count

scanned mode의 경우 반복되는 값에 대한 count 최대 값을 지정한다.

## disconnect

disable 채널에 대한 연결 유지 유무를 설정한다. 기본 설정으로 disable 채널들은 조건을 만족할 때 그룹에 대한 Archive를 비활성화 시키지만 자신은 Channel Access를 통해 연결을 유지하고 있다. 만약 disconnect 설정이 되어있다면 이러한 disable 채널들에 대한 Channel Access연결 또한 해제 한다.

## group

다수의 채널을 그룹으로 묶어 관리할 수 있으며 2개의 하위 태그를 가지고 있다.

- name: 그룹 이름을 지정한다.
- channel: 다음 5개의 하위 태그를 가지고 있다.
  - name: 채널 이름을 지정한다.
  - period: 채널 값을 저장하는 주기를 지정한다.
  - scan: period에 설정된 시간 주기로 scan mode로 작동한다.
  - monitor: monitor mode로 작동한다. period 보다 빠른 빈도로 값이 변하면 무시한다.
  - disable: disable 태그가 설정된 channel값이 0보다 크면 같은 그룹에 속해있는 channel들의 archive가 비활성 된다.

## 3.2 Starting and Stopping

ArchiveEngine을 시작하기 위해 다음과 같이 실행한다.

```
ctrluser@ctrluser:~\$ ArchiveEngine engineconfig.xml ./index
07/23/2014 13:39:57 Starting Engine with configuration file engineconfig.xml, index ./index
07/23/2014 13:39:57 Creating ChannelAccess Context.
07/23/2014 13:39:57 Engine starts.
07/23/2014 13:39:57 Starting EngineServer
07/23/2014 13:39:57
-----
Engine Running. Stop via http://localhost:4812/stop
-----
07/23/2014 13:39:57 HTTPServer thread 0x01D24680 running.
```

Engine이 시작되면 중복 실행되는 것을 방지하기 위해 lock파일(`archive_active.lck`)이 자동으로 생성된다. 따라서 `archive_active.lck`가 존재하는 상태에서는 Engine을 시작할 수 없게된다. Engine이 중지되면 lock 파일은 자동으로 삭제되는데, 만약 시스템 오류등으로 제대로 중지되지 않을 경우 lock파일은 삭제되지 않는다. 이 경우 강제적으로 삭제하면 다시 시작할 수 있게 된다.

Engine을 중지하기 위해서는 키보드의 “CTRL-C”를 누르거나 Engine을 시작할 때 나오는

웹 주소 “http://localhost:4812/stop” 명령을 통해 중지할 수 있다.

여러 Engine을 동시에 시작하고자 할 경우에는 다음 2가지 사항에 따라 진행한다.

- Engine하나당 하나의 lock파일만 생성되므로 각각의 폴더를 만들어서 실행시킨다.
- 웹 서버의 TCP포트 중복을 막기 위해 “-p <port>” 옵션을 이용하여 서로 다른 포트를 설정한다.

웹 서버의 TCP포트는 기본 4812포트를 사용하며 “-p” 옵션을 통해 다음과 같이 변경이 가능하다.

```
ctrluser@ctrluser:~\$ ArchiveEngine -p 5001 engineconfig.xml ./index
07/23/2014 14:41:44 Starting Engine with configuration file engineconfig.xml, index index
07/23/2014 14:41:44 Creating ChannelAccess Context.
07/23/2014 14:41:44 Engine starts.
07/23/2014 14:41:44 Starting EngineServer
07/23/2014 14:41:44
-----
Engine Running. Stop via http://localhost:5001/stop
-----
07/23/2014 14:41:44 HTTPServer thread 0x00F419A0 running.
```

## 4 Data Retrieval

Archive Engine에 의해 파일로 저장된 데이터를 읽는 방법은 크게 2가지 방법이 있다.

- Archive Export  
데이터 파일에 직접 접근하여 텍스트로 출력한다.
- Archive Data Server  
CSS, Java Archvie Viewer와 같은 Client Tool을 이용하여 그래프로 출력한다.

### 4.1 Archive Export

Archive Export는 명령어 라인 툴로서 별도의 설정이 필요없이 데이터 파일에 직접 접근하여 원하는 데이터를 텍스트로 출력해 준다. 기본 적인 사용방법은 다음과 같다.

USAGE: ArchiveExport [Options] <index file> {channel}

Options:	
-verbose	Verbose mode
-match <reg. exp.>	Channel name pattern
-list	List all channels
-info	Time-range info on channels
-start <time>	Format: "mm/dd/yyyy[ hh:mm:ss[.nano-secs]]"
-end <time>	(exclusive)
-text	Include text column for status/severity (default)
-no_text	Exclude text column for status/severity
-output <file>	Output to file
-plotbin <seconds>	Bin the raw data for plotting
-average <seconds>	average values
-linear <seconds>	Interpolate values linearly
-format <decimal engineering exponential>	Use specific format for numbers

-precision <int>	Precision of numbers
-gnuplot	Generate GNUPlot command file
-Gnuplot	Generate GNUPlot output for Image
-raw_time	Include columns for EPICS time stamp
-millisecs	Truncate time to millisecs in spreadsheet dump.

Archive Export는 출력 형식에 따라 다양한 설정들을 지원한다. 예를 들어 채널 값을 텍스트로 출력하려면 다음과 같이 하면 된다.

```
ctrluser@ctrluser:~\$ ArchiveExport -l index
ab:HUM
ab:TEM
ctrluser@ctrluser:~/ ArchiveExport -output data.txt index ab:HUM
ctrluser@ctrluser:~/ cat data.txt
...
...
...
06/19/2014 15:59:20.871534536 16995
06/19/2014 15:59:21.871677190 17347
06/19/2014 15:59:22.871693851 17140
06/19/2014 15:59:23.871531523 17099
06/19/2014 15:59:24.871585182 17207
06/19/2014 15:59:25.871533847 17247
06/19/2014 15:59:26.871683500 17197
06/19/2014 15:59:27.871794155 17261
06/19/2014 15:59:28.871528834 17078
07/23/2014 14:27:17.950156693 #N/A Archive_Off
07/23/2014 14:42:45.201544627 #N/A Archive_Off
```

특정 시간에 대한 값을 출력하고자 한다면 다음과 같이 하면 된다.

```
ctrluser@ctrluser:~\$ ArchiveExport -output data.txt -s "06/19/2014 15:50:00" -e "06/19/2014
15:51:00" index ab:HUM
ctrluser@ctrluser:~\$ cat data.txt
# Time ab:HUM []
06/19/2014 15:49:59.871533064 16549
06/19/2014 15:50:00.871712716 16785
06/19/2014 15:50:01.871538388 16933
06/19/2014 15:50:02.871716040 17092
06/19/2014 15:50:03.871572710 17030
06/19/2014 15:50:04.871546374 16833
06/19/2014 15:50:06.871567697 16833 Repeat 1
06/19/2014 15:50:06.871567697 16649
06/19/2014 15:50:07.871731350 16869
06/19/2014 15:50:08.871589020 16880
06/19/2014 15:50:09.871528686 16966
06/19/2014 15:50:10.871585345 16904
06/19/2014 15:50:11.871530010 16711
06/19/2014 15:50:12.871751659 16399
06/19/2014 15:50:13.871588331 16730
06/19/2014 15:50:15.871706648 16730 Repeat 1
06/19/2014 15:50:15.871706648 16964
06/19/2014 15:50:16.871692311 16559
06/19/2014 15:50:17.871672975 16588
06/19/2014 15:50:18.871730633 16819
06/19/2014 15:50:19.871536307 16585
06/19/2014 15:50:20.871588966 17045
06/19/2014 15:50:21.871520633 16935
06/19/2014 15:50:22.871718283 16759
06/19/2014 15:50:23.871734944 16640
06/19/2014 15:50:24.871524619 16690
06/19/2014 15:50:25.871732268 16719
```

```

06/19/2014 15:50:26.871535942 16435
06/19/2014 15:50:27.871726593 16599
06/19/2014 15:50:28.871696257 16861
06/19/2014 15:50:29.871511930 16642
06/19/2014 15:50:30.871709581 16592
06/19/2014 15:50:31.871568251 16464
06/19/2014 15:50:32.871740903 16680
06/19/2014 15:50:33.871731566 16516
06/19/2014 15:50:34.871524240 16657
06/19/2014 15:50:35.871685893 16626
06/19/2014 15:50:36.871538564 16761
06/19/2014 15:50:37.871751213 16457
06/19/2014 15:50:38.871589885 16826
06/19/2014 15:50:39.871542550 16421
06/19/2014 15:50:40.871708202 16697
06/19/2014 15:50:41.871565873 16688
06/19/2014 15:50:42.871551536 16214
06/19/2014 15:50:43.871581196 16733
06/19/2014 15:50:44.871559860 16547
06/19/2014 15:50:45.871697514 16683
06/19/2014 15:50:46.871824168 16373
06/19/2014 15:50:47.871528848 16626
06/19/2014 15:50:48.871720499 16716
06/19/2014 15:50:49.871531172 16669
06/19/2014 15:50:50.871694825 16561
06/19/2014 15:50:51.871581494 16147
06/19/2014 15:50:52.871530159 16378
06/19/2014 15:50:53.871723809 16516
06/19/2014 15:50:54.871543483 16609
06/19/2014 15:50:55.871693136 16538
06/19/2014 15:50:56.871539807 16430
06/19/2014 15:50:57.871711459 16509
06/19/2014 15:50:58.871699122 16633
06/19/2014 15:50:59.871561792 16395

```

## 4.2 Archive Data Server

Data Server를 사용하기 위해서는 몇가지 설정이 필요하다.

### 설치

ArchiveDataServer를 ArchiveDataServer.cgi로 변경한 후 /var/www/html/archive/cgi/로 옮긴다. serverconfig.dtd 파일을 만든 후 아래 내용을 추가한다.

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- DTD for the XML-RPC Data Server Configuration -->
<!ELEMENT serverconfig (archive+)>
<!ELEMENT archive (key,name,path)>
<!ELEMENT key (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT path (#PCDATA)>

```

서버 설정을 위해 serverconfig.xml 파일을 만든 후 다음 3가지 내용을 추가한다.

- key: 각각의 Archive에 대하여 독립적인 키값을 설정한다.
- name: Archive 이름을 설정한다.



- path: index파일이 위치한 경로를 설정한다.

## 설정

관리자 권한으로 /etc/apache2/apache2.conf 파일에 아래 내용을 추가한 후 서버를 재시작한다.

```
ctrluser@ctrluser:~\$ sudo vi /etc/atache2/apache2.conf
AddHandler cgi-script .cgi

<Directory /var/www/html/archive>
Order Allow,Deny
Allow from All
</Directory>

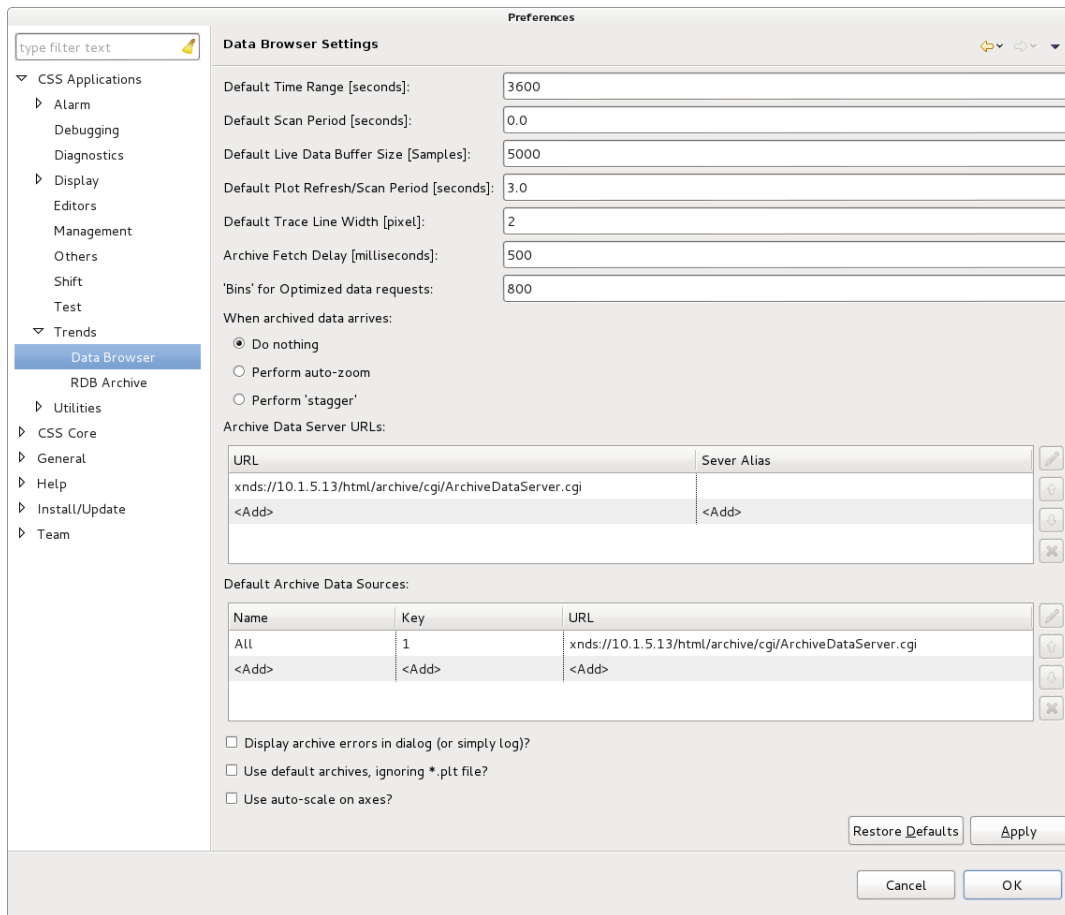
<Directory /var/www/html/archive>
SetEnv "EPICS_TS_MIN_WEST" "300"
SetEnv "LD_LIBRARY_PATH" "/lib:/usr/lib:/usr/local/lib:/home/ctrluser/epics/base-3.14.12.4/lib/
    linux-x86_64:/home/ctrluser/epics/extensions_for_3.14.12.4/lib/linux-x86_64:/usr/lib:/usr/
    local/lib"
SetEnv "SERVERCONFIG" "/var/www/html/archive/cgi/serverconfig.xml"
#PerlHandler Apache::Registry
#PerlSendHeader On
Options +ExecCGI
</Directory>

ctrluser@ctrluser:~\$ /etc/init.d/apache2 restart
```

## CSS 테스트

데이터 서버 테스트를 위해 CSS를 실행 하여 다음과 같이 설정한다.

- Edit-Preferences를 실행.
- CSS Applications-Trends-Data Browser 선택.
- Archive Data Server URLs에 다음의 Data Server 주소 추가.
  - xnds://10.1.5.13/html/archive/cgi/ArchiveDataServer.cgi
- File-Restart CSS 실행
- CSS-Trends-Data Browser 실행
- Archive Search URL을 설정한 Data Server로 선택
- Pattern에 PV Name또는 \*를 입력한 후 Search 클릭
- 검색된 PV Nmae위에 마우스 오른쪽 버튼을 눌러 Process Variable-Data Broser 선택
- Data Browser에서 Value Axes를 조정하여 Y축 범위를 PV 값에 맞게 조절한다.



**Figure 1** Data Browser Setting

## Bibliography

- [1] Channel Archiver Manual, 2014.  
<http://ics-web.sns.ornl.gov/kasemir/archiver/manual.pdf> (accessed July 22, 2014).
- [2] Channel Archiver Home, 2014.  
<http://ics-web.sns.ornl.gov/kasemir/archiver> (accessed July 22, 2014).
- [3] Channel Archiver Source, 2014.  
<http://epicschanarch.cvs.sourceforge.net/viewvc/epicschanarch/> (accessed July 22, 2014).

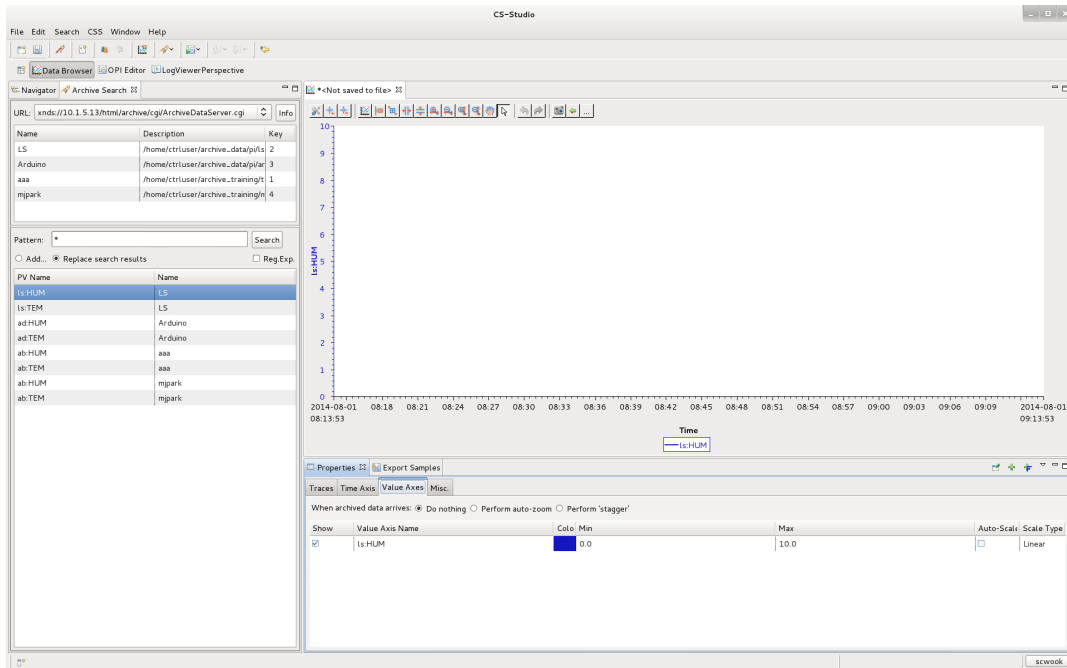


Figure 2 Data Browser

## Appendix A

```

1  !/bin/bash
2  # Shell : epics_chanarch.sh
3  # Author : Jeong Han Lee
4  # email : jhlee@ibs.re.kr
5  # Date : Tuesday, February 4 23:29:02 KST 2014
6  # version : 0.0.1
7  #
8  #
9  # * This script may give a little room to install again-and-again
10 # Classic EPICS Channel Archiver located at
11 # http://sourceforge.net/projects/epicschanarch/
12 #
13 # I hope it would reduce our time to concentrate real works
14 # instead of painful modification of unmaintained source codes.
15 #
16 # * Must install the following packages via root or sudo
17 # before running this script
18 #
19 # For Debian Wheezy,
20 #
21 # aptitude install libxmlrpc-c++4-dev libxerces-c-dev libcurl3-dev
22 #
23 # Of course, EPICS base, and its extension must be installed
24 #
25 #
26 #
27 # - 0.0.1 Wednesday, February 5 00:47:14 KST 2014, jhlee

```

```

28 #          * created
29 #
30
31
32
33
34 # cq    : quiet
35 # c      : continue getting a partially-downloaded file. So it allows us not to re-download an
            existent file
36 #
37 wget_options="wget -c"
38 # xzf    : quiet
39 # xzfv   : verbose
40 tar_command="tar xzf"
41
42 # aptitude install libxmlrpc-c++4-dev libxerces-c-dev libcurl3-dev
43
44 make_epicschanarch()
45 {
46
47     chanarch_dir=$1
48     cd ${chanarch_dir}
49     echo $PWD
50     echo ""
51     make_cfg="${chanarch_dir}/make.cfg"
52     sed -i~ -e 's/LOCAL=/usr/local/LOCAL=/usr//g' ${make_cfg}
53
54
55     # modify Tools in order to compile them correctly
56     # Tuesday, February 4 18:42:16 KST 2014, jhlee
57
58     tools_dir=${chanarch_dir}/Tools/
59     sed -i~ '$ a#include <stdio>\n#include <stdlib>\n#include <stdarg>' ${tools_dir}/
        ToolsConfig.h
60     sed -i~ '18istatic int sort_compare(const int &a, const int &b)\n{\n return b-a;\n}\n\nstatic
        const char *toString(const int &i)\n{\n static char txt[10];\n sprintf(txt, "%d", i);\n
        return txt;\n }\n' ${tools_dir}/AVLTree.h
61     sed -i~ '18,23d' ${tools_dir}/AVLTreeTest.cpp
62     sed -i~ '5,6d' ${tools_dir}/AVLTreeTest.cpp
63
64     sed -i -e 's/#include <expat.h>/#include <xmlparse.h>/g' ${tools_dir}/FUX.cpp
65
66
67     # modify LibIO
68     # error: ~@~XLONG_MAX~@~Y was not declared in this scope >> BinValue.cpp
69     # ./DataFile.h:131:5: error: ~@~XDataHeaderIterator~@~Y does not name a type >> DataFile.cpp
70     #
71     libIO_dir=${chanarch_dir}/LibIO/
72     sed -i~ '7i#include <climits>\n#include "BinChannelIterator.h"' ${libIO_dir}/BinValue.h
73
74
75     # modify Storage
76     # ../NameHash.cpp:4:23: warning: extra tokens at end of #include directive [enabled by default
        ]
77
78     storage_dir=${chanarch_dir}/Storage/
79     sed -i~ -e 's/.h>./h>/g' ${storage_dir}/NameHash.cpp
80
81     # modify XMLRPCServer
82     # Cannot run or decode 'xmlrpc-c-config'
83     # ./main_standalone.cpp:81:101: warning: deprecated conversion from string constant to ~@~
        Xchar*~@~Y [-Wwrite-strings]

```

```

84
85 xmlrpcserver_dir=${chanarch_dir}/XMLRPCServer/
86
87 sed -i~ -e 's/1.0/1./g' ${xmlrpcserver_dir}/xmlrpc-config-wrapper
88 sed -i~ -e 's/"archiver./(char*)"archiver./g' ${xmlrpcserver_dir}/main_standalone.cpp
89 sed -i -e 's/"S:/(char*)"S:/g' ${xmlrpcserver_dir}/main_standalone.cpp
90 sed -i -e 's/"A:/(char*)"A:/g' ${xmlrpcserver_dir}/main_standalone.cpp
91 sed -i -e 's/"Get /(char*)"Get /g' ${xmlrpcserver_dir}/main_standalone.cpp
92
93 # Engine
94 # ../hammer.cpp:28:17: error: ~@~Xstderr~@~Y was not declared in this scope
95 # ../hammer.cpp:29:58: error: ~@~Xfprintf~@~Y was not declared in this scope
96 engine_dir=${chanarch_dir}/Engine/
97 sed -i~ -e 's/#include <string>/#include <cstdio>\n#include <cstdlib>\n#include <cstring>\n#
include <iostream>/g' ${engine_dir}/hammer.cpp
98
99
100 cd ${chanarch_dir}
101 make
102 }
103
104 chanarch_name="ChannelArchiver"
105 EXT_SRC=${EPICS_EXTENSIONS}/src
106
107
108 chanarch_filename=epicschanarch.tar.gz
109 cd ${EXT_SRC}
110
111 $wget_options -O ${chanarch_filename} http://epicschanarch.cvs.sourceforge.net/viewvc/
epicschanarch/?view=tar
112 $tar_command ${chanarch_filename} --strip-components=1
113
114 #cd ${EXT_SRC}/${chanarch_name}
115
116 make_epicschanarch "${EXT_SRC}/${chanarch_name}"

```

**Listing A.1** chanarch.sh

## Appendix B

```
1  #!/usr/bin/perl
2
3  $configFile = "engineconfig.xml";
4
5  $dirName = ".";
6
7  -e $configFile || die "$configFile is not exist. Please use 'makeArchiveEngineConfig.pl'\n";
8
9  @psl = `ps aux | grep ArchiveEngine`;
10
11  $basePort = 5001;
12  $machString = "-p $basePort";
13
14  foreach $num (0..$#psl)
15  {
16      foreach $idx (0..$#psl)
17      {
18          if( $psl[$idx] =~ /$machString/ )
19          {
20              $basePort++;
21              $machString = "-p $basePort";
22              last;
23          }
24      }
25  }
26
27  close FH;
```

**Listing B.1** ArchiveEngineStart.pl

## Appendix C

```
1
2 #!/usr/bin/perl
3
4 open ( fileHandle, ">engineconfig.dtd" );
5
6 print fileHandle <<END_DTD;
7 <?xml version="1.0" encoding="UTF-8"?>
8 <!-- DTD for the ArchiveEngine Configuration -->
9 <!-- Note that we do not allow empty configurations: -->
10 <!-- Each config. must contain at least one group, -->
11 <!-- and each group must contain at least 1 channel. -->
12 <!ELEMENT engineconfig ((write_period|get_threshold|
13     file_size|ignored_future|
14     buffer_reserve|
15     max_repeat_count|disconnect)*,
16     group+)>
17 <!ELEMENT group (name,channel+)>
18 <!ELEMENT channel (name,period,(scan|monitor),disable?)>
19 <!ELEMENT write_period (#PCDATA)><!-- int seconds -->
20 <!ELEMENT get_threshold (#PCDATA)><!-- int seconds -->
21 <!ELEMENT file_size (#PCDATA)><!-- MB -->
22 <!ELEMENT ignored_future (#PCDATA)><!-- double hours -->
23 <!ELEMENT buffer_reserve (#PCDATA)><!-- int times -->
24 <!ELEMENT max_repeat_count (#PCDATA)><!-- int times -->
25 <!ELEMENT disconnect EMPTY>
26 <!ELEMENT name (#PCDATA)>
27 <!ELEMENT period (#PCDATA)><!-- double seconds -->
28 <!ELEMENT scan EMPTY>
29 <!ELEMENT monitor EMPTY>
30 <!ELEMENT disable EMPTY>
31 END_DTD
32
33 close( fileHandle );
34
35 open( fileHandle, ">engineconfig.xml" ) || die "Failed opening.\n";
36
37 print fileHandle <<END_XML;
38 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
39 <!DOCTYPE engineconfig SYSTEM "engineconfig.dtd">
40 <engineconfig>
41     <write_period>30 sec</write_period>
42     <get_threshold>10 sec</get_threshold>
43     <file_size>30</file_size>
44     <ignored_future>1 hour</ignored_future>
45     <buffer_reserve>3</buffer_reserve>
46     <max_repeat_count>120</max_repeat_count>
47     <group>
48         <name>AB</name>
49         <channel>
50             <name>Pi:bi14</name>
51             <period>1 sec</period><scan/>
52         </channel>
53     </group>
54 </engineconfig>
55 END_XML
56
57 close( fileHandle );
```

**Listing C.1** makeArchiveEngineConfig.pl

## Appendix D

```
1  !/usr/bin/python3.2
2
3  import os
4  import sys
5  import xml.etree.ElementTree as etree
6
7  def add_index(root, key, name, path):
8      if not os.path.exists(path):
9          print("Cannot find the \'index\' file:")
10         print(path)
11         sys.exit()
12
13         e_archive = root.findall("archive")
14         for e in e_archive:
15             e_name = e.find("name")
16             if e_name.text == name:
17                 print("%s is already in use, Please select other name" % name)
18                 sys.exit()
19
20         e_archive = etree.Element("archive")
21         etree.SubElement(e_archive, "key").text = key
22         etree.SubElement(e_archive, "name").text = name
23         etree.SubElement(e_archive, "path").text = path
24
25         root.append(e_archive)
26
27  def del_index(root, name):
28      e_archive = root.findall("archive")
29      exist_name = 0
30      for e in e_archive:
31          e_name = e.find("name")
32          if e_name.text == name:
33              root.remove(e)
34              print("%s is deleted" % name)
35              exist_name = 1
36
37      if not exist_name:
38          print("Cannot find the %s" % name)
39          sys.exit()
40
41  def find_available_key(root):
42      e_archive = root.findall("archive")
43      available_key = 1
44      for num in e_archive:
45          for e in e_archive:
46              e_key = e.find("key")
47              if int(e_key.text) == available_key:
48                  available_key += 1
49              break
50
51      return available_key
52
53  def print_usage_error():
54      print("\nUSAGE: DataServerConfig [Options] <name>\n")
55      print("Options:")
56      print("  -a    Add index")
57      print("  -d    Delete index")
58      print("  -l    Show index list")
59
```



```

60 def print_archive_list(root):
61     e_archive = root.findall("archive")
62     for e in e_archive:
63         print(e[0].text, e[1].text, e[2].text)
64
65 def indent(root, level=0):
66     i = "\n" + level*" "
67     if len(root):
68         if not root.text or not root.text.strip():
69             root.text = i + " "
70         if not root.tail or not root.tail.strip():
71             root.tail = i
72         for root in root:
73             indent(root, level+1)
74         if not root.tail or not root.tail.strip():
75             root.tail = i
76     else:
77         if level and (not root.tail or not root.tail.strip()):
78             root.tail = i
79
80 #####
81 if len(sys.argv) < 2 or len(sys.argv) > 3:
82     print_usage_error()
83     sys.exit()
84
85 file_name = "serverconfig.xml"
86 file_path = "/var/www/html/archive/cgi/" + file_name
87
88 if not os.path.exists(file_path):
89     print("Cannot find the \'%s\' file:" % config_name)
90     sys.exit()
91
92 root = etree.parse(file_path).getroot()
93
94 option = sys.argv[1]
95 if len(sys.argv) == 2:
96     if option == "-l":
97         print_archive_list(root)
98         sys.exit()
99     else:
100         print_usage_error()
101         sys.exit()
102 elif len(sys.argv) == 3:
103     key_val = str(find_available_key(root))
104     name_val = sys.argv[2]
105     path_val = os.getcwd() + "/index"
106     if option == "-a":
107         add_index(root, key_val, name_val, path_val)
108     elif option == "-d":
109         del_index(root, name_val)
110     else:
111         print_usage_error()
112         sys.exit()
113 else:
114     print_usage_error()
115     sys.exit()
116
117 indent(root)
118 etree.ElementTree(root).write(file_path)
119
120 with open(file_path, "r") as xml_file:
121     data = xml_file.read()

```

```
122
123 with open(file_path, "w") as config_file:
124     config_file.write('<?xml version="1.0" encoding="UTF-8"?>\n<!DOCTYPE serverconfig SYSTEM "
125     serverconfig.dtd">\n')
126     config_file.write(data)
```

**Listing D.1** DataServerConfig.py