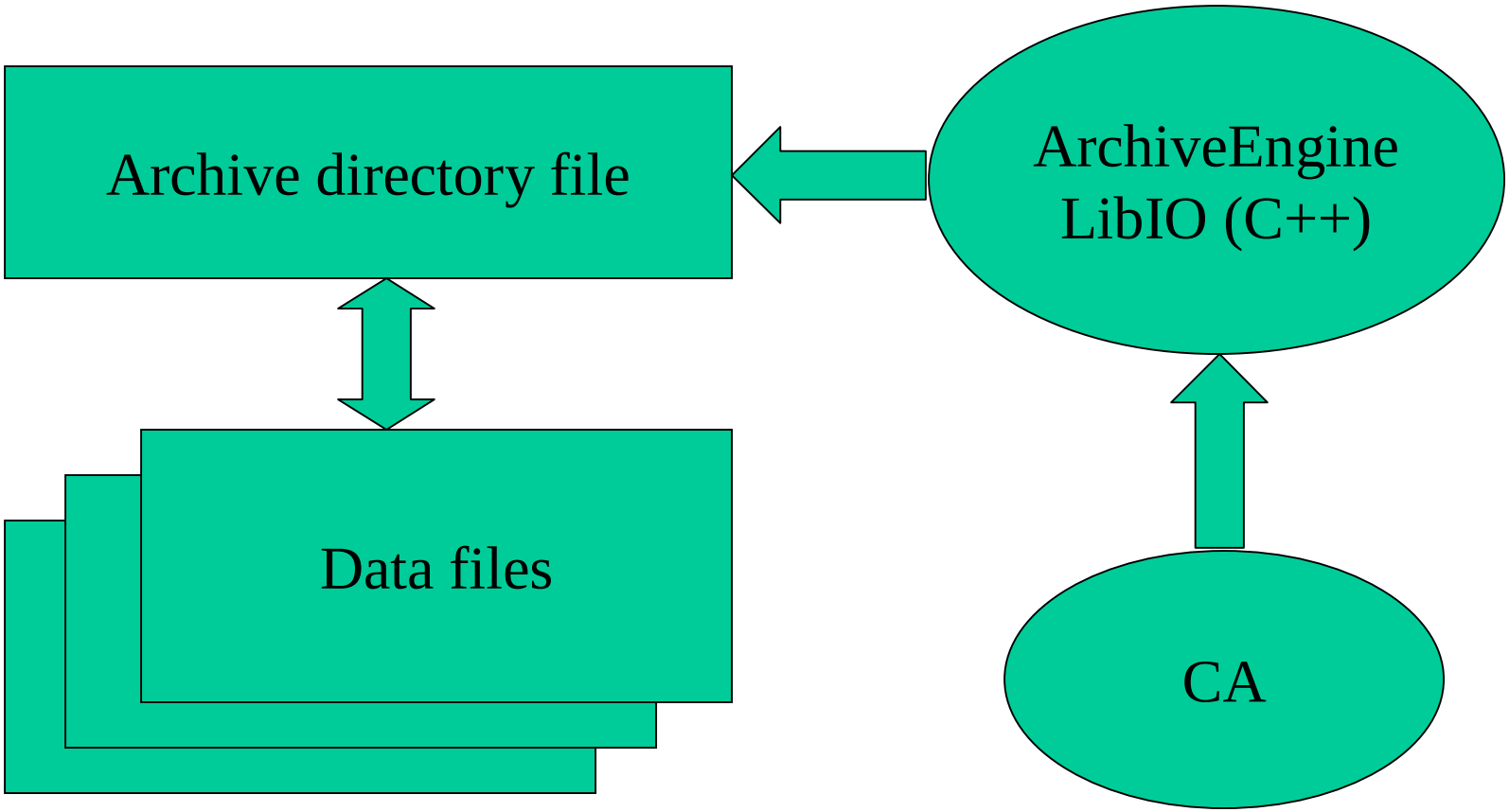
DØ Channel Archiver Tutorial

V.Sirotenko, 4/4/2001

Idea of ChannelArchiver

All channels are written into a single file. The list of channel names, the first and last position for each channel etc. is kept in a disk-based hash table. Furthermore, a binary format is used to save space and time.
The disk-based hash-table is the "directory" file.
As for a the data, one file per day (default) is used.
(A data file will be updated until all the buffers in there are filled. The filename mostly reflects the date of creation, not the time regime of samples therein)
Advantage: This format allows archiving of up to 10000 values per sec on a 450 Mhz machine, 100baseT network. Fast retrieval to the most recent values is accomplished via the hash table and a pointer to the last entries.
Disadvantage: The combination of directory- and data-files must not be disturbed!
Under no circumstances must any single data file be removed or altered, since this will break the interlinked sequence of samples.
This is not uncommon for database systems, but unfortunately the data management tools for the ChannelArchiver are limited for the time being.

Idea of ChannelArchiver (cont)



Time Stamps:

Confusing point:

"Setpoint" channels on an IOC,
which are usually passive and only changed from an operator interface, initially have no time stamp at all until they are changed by an operator
-> Warnings "invalid time stamp 00:00:00", value is not archived

A setpoint that was changed a week ago still has that time stamp
-> the sample is put in a data file with a week old time stamp

Files are updated until the buffers in there are full
-> "old" files are still changed

Channel Archiver Toolkit

ArchiveEngine : logger (Linux,NT)
ArchiveManager : archive monitoring program (Linux,NT)

Tools: Python scripts
- ArchiveInfo.py, ChannelInfo.py, ListChannels.py, ListBrowser.py,Plot.py,XYPlot.py,dump.py,copy.py

Tcl/Tk scripts
- ListChannels.tcl, dump.tcl, lastTimes.tcl, lastValues.tcl, spreadsheetExport.tcl

CGIExportTools (run from the Web)
ChannelArchiver I/O Library (C++)
WinBrowser (NT)

ChannelArchiver I/O Library Classes

- Archive
- ArchiveI
- BinArchive
- ChannelI
- ChannelIterator
- ChannelIteratorI
- CtrlInfoI
- ExpandingValueIteratorI
- InfoFilterValueIteratorI
- LinInterpolValueIteratorI
- MultiArchive
- RawValueI
- ValueI
- ValueIterator
- ValueIteratorI

Content of the archive:

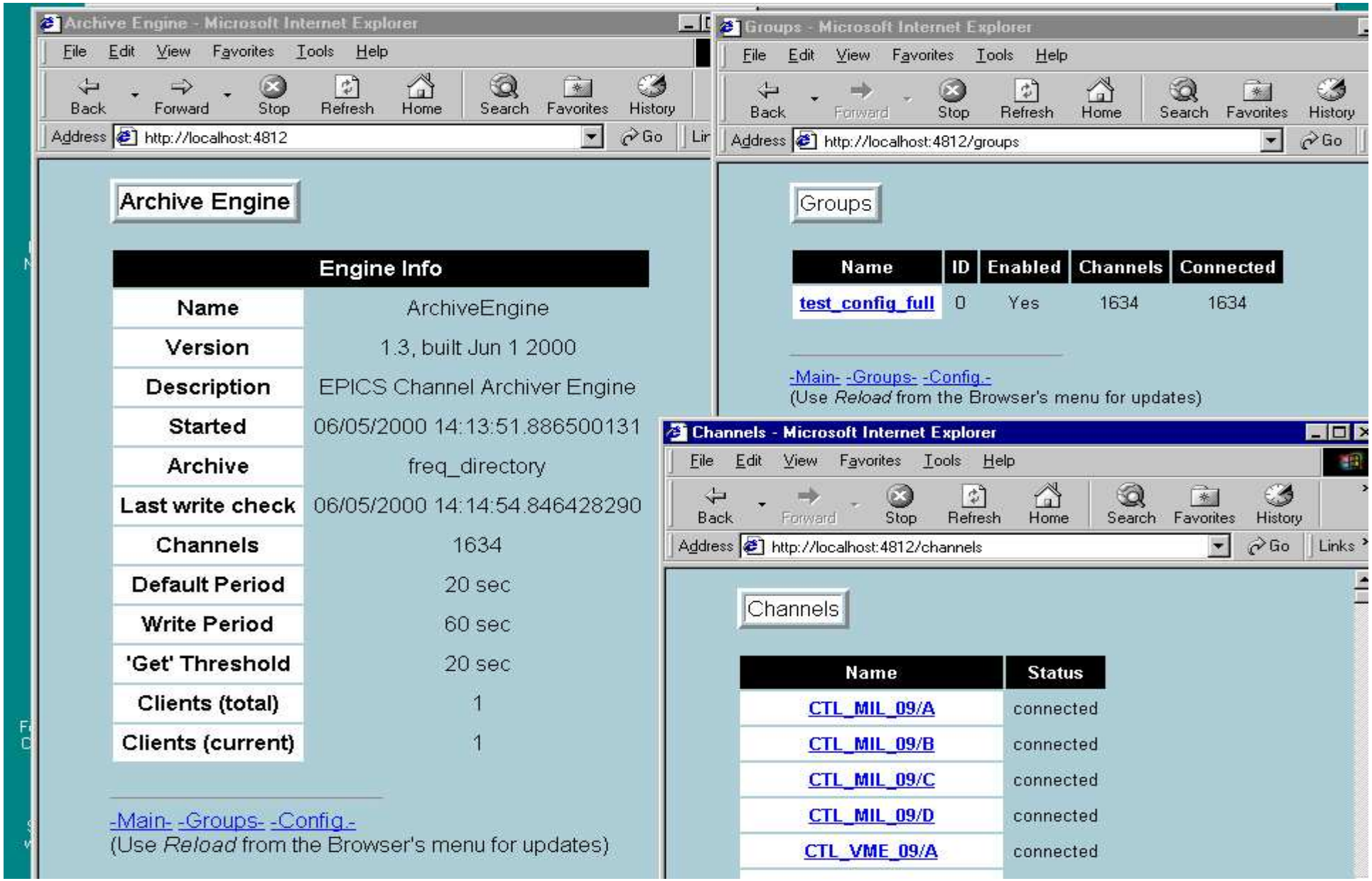
```
CALC_CMCA_02/TE00BR_TIMEDBR_TIME_DOUBLE/2000/06/07 10:44:18.716242700
. . . . . .
CTL_MIL_11/C    DBR_TIME_LONG   2000/06/07 10:44:20.000000000
CTL_MIL_11/D    - no values -
CTL_PROC_09/CPU DBR_TIME_DOUBLE 2000/06/07 10:44:28.089904210
```

Timestamps: come directly from IOC

File size: 82 Mb for 1600 PV's /day/minute (about 30 bytes per value)
It saves each value in its native type (double, short, int, string, array, whatever)
plus additional information:

* each time stamp
* each status/severity
* control infomation:
  - alarm limits, display limits, control limits, eng. units
  - enumeration strings for enumerated data

ArchiveEngine Web Interface



ArchiveEngine Web Interface (cont)

What it is for:

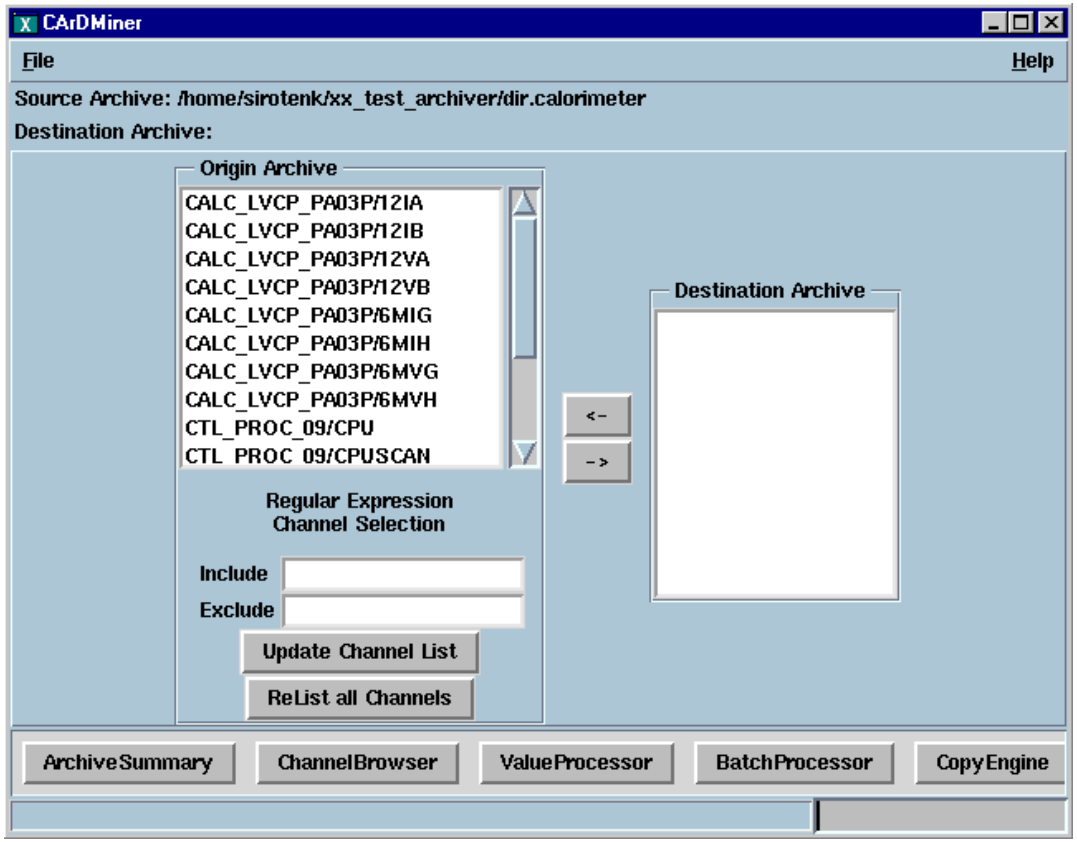
- interface to control status of the running ArchiveEngine,
  (channel connections, IOC names, scanning rates etc.)
- add groups of channels or individual channels on fly,
  disable or enable them, change get/monitor access methods
- can event stop archiver

ArchiveManager Tool

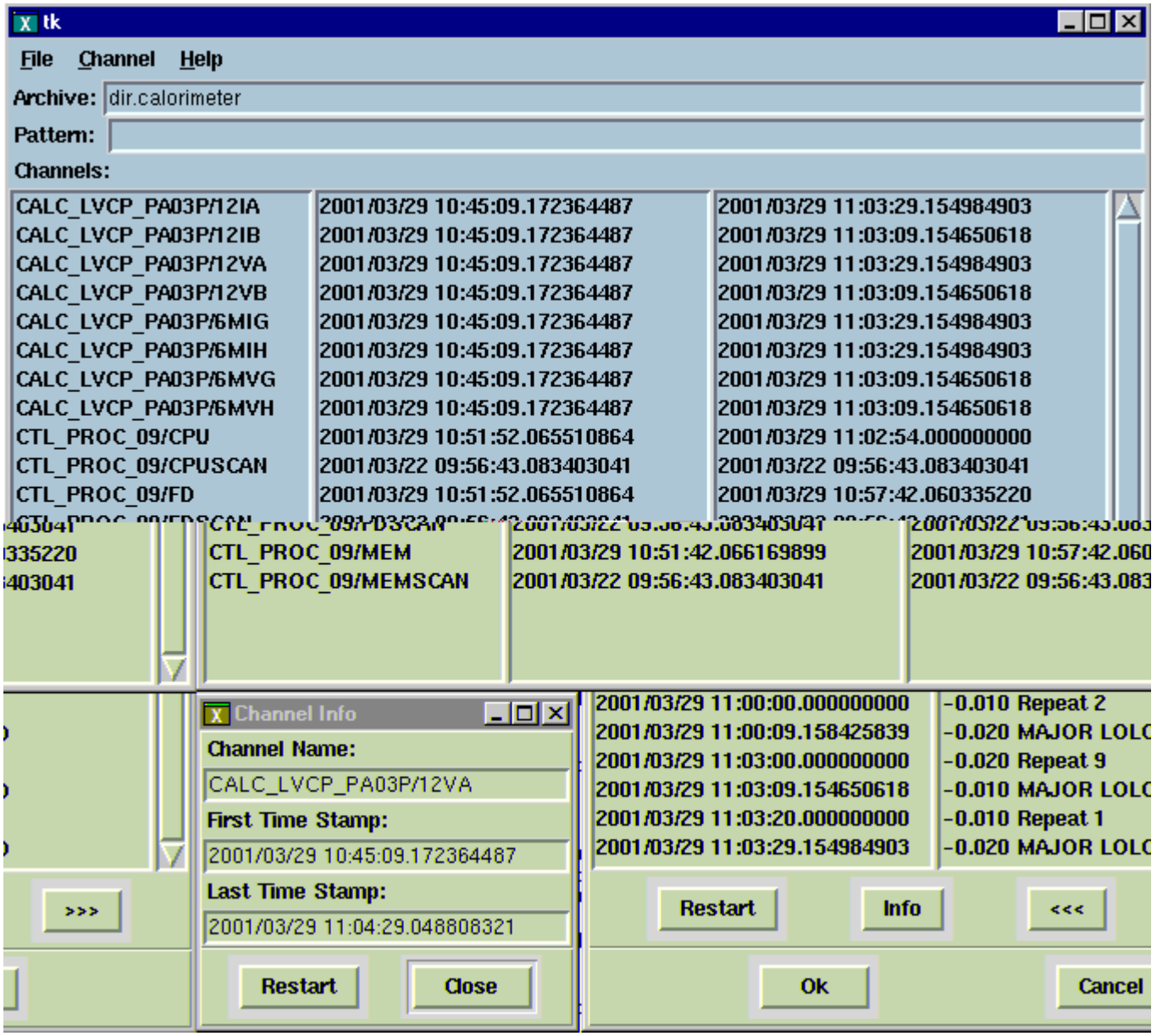
```
Linux:onl->ArchiveManager
Archive Manager version 1.6.3, built Mar 19 2001
USAGE:  ArchiveManager [Options] <archive>
Options:
        -info                   Show archive information
        -test                   Test archive for errors
        -channel <channel>          Specify channel name
        -match <regular expression>    List matching channels
        -Match <regular expression>    Dump values for matching channels
        -start <time>               Start time as mm/dd/yyyy hh:mm:ss[.nano-secs]
        -end <time>                 End time (exclusive)
        -xport <new archive>        export data into new archive
        -repeat_limit <seconds>     remove 'repeat' entries beyond limit (export)
        -FileSize <days>            Days per binary data file (export, binary file format detail)
        -headers <channel>          show headers for channel
        -Output <channel>           output ASCII dump for channel
        -Input <ascii file>         read ASCII dump for channel into archive
        -Compare <target archive>   Compare with target archive
        -Seek                   Seek test (use with -start)
```

ArchiveManager Tool (cont)

There is python GUI which is a wrapper for ArchiveManager: CArDMiner.py. Convenient, but slower and was not evaluated for use in D0 yet



ArchiveInfo.py



Dump.py

Linux:onl->dump.py dir.20010323-152753 CALC_CMCP_PA02/FTB

First Sample: 2001/03/23 15:27:59.741485425
Last Sample: 2001/03/27 08:37:09.209401125
2001/03/23 15:27:59.741485425 -26.568 MAJOR LOLO
2001/03/23 15:28:39.740669035 -26.080 MAJOR LOLO
2001/03/23 15:28:40.000000000 -26.080 Repeat 1
2001/03/23 15:28:59.740760767 -27.056 MAJOR LOLO
2001/03/23 15:29:00.000000000 -27.056 Repeat 1
2001/03/23 15:29:19.739890019 -26.568 MAJOR LOLO
2001/03/23 15:30:20.000000000 -26.568 Repeat 4
2001/03/23 15:30:39.739221967 -26.080 MAJOR LOLO

spreadsheetExport.tcl

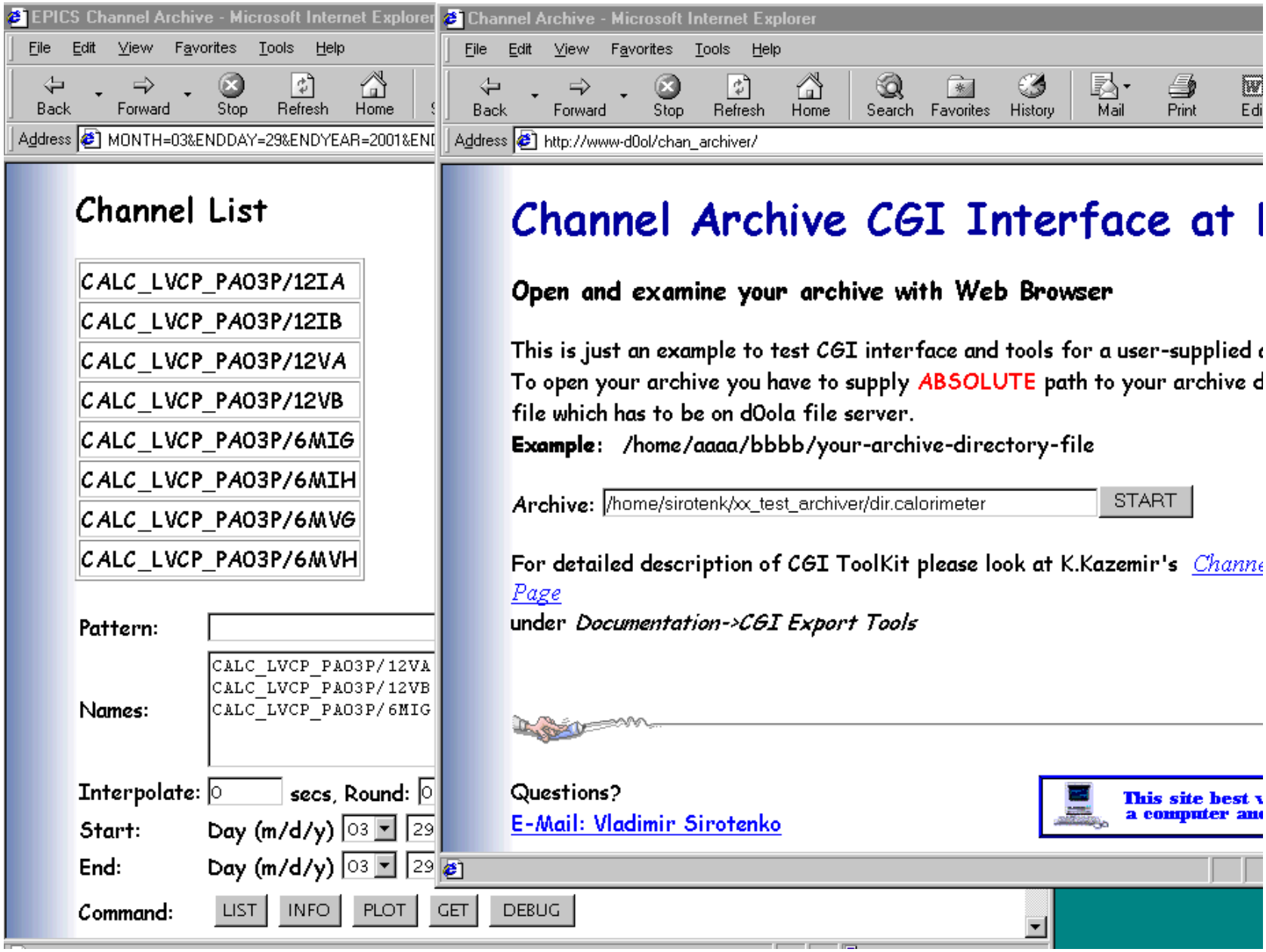
Allows to export archived information in spreadsheet format for later use in Excel

> spreadsheetExport_tcl <archive_file> <start-time> <end-time> <varname1> [<varname2> ..]

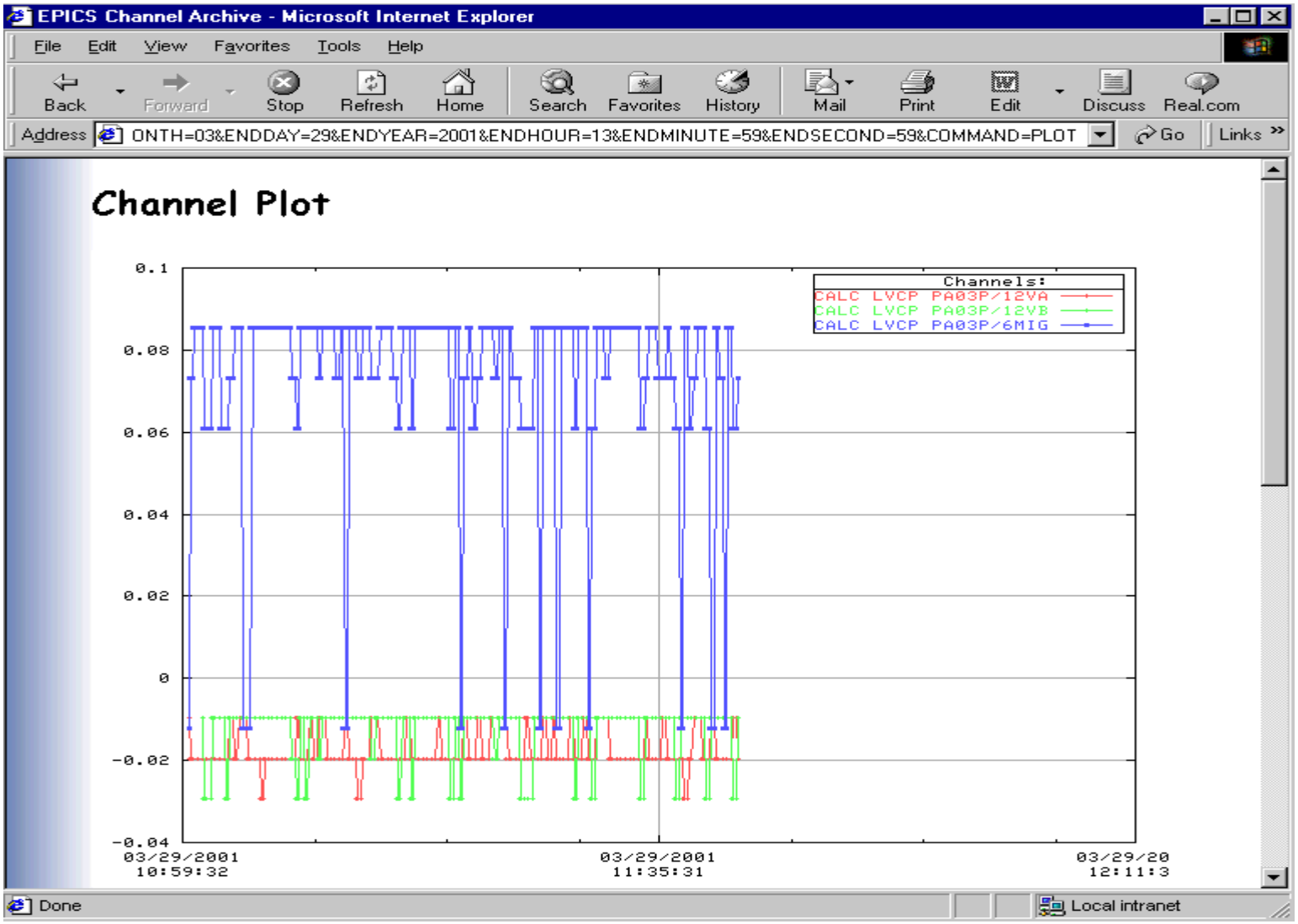
```
Linux:onl->spreadsheetExport_tcl dir.20010329-121532 "2001/04/03 23:00:00" end
SMT_FINGER_IFN225/RLVH

Time                        SMT_FINGER_IFN225/RLVH
2001/04/03 23:00:00     0.051293814
2001/04/03 23:00:00.979028488   0.051318228
2001/04/03 23:00:01.000000000   0.051318228
2001/04/03 23:00:01.979133488   0.0512694
2001/04/03 23:00:02.000000000   0.0512694
2001/04/03 23:00:02.979238488   0.051293814
```

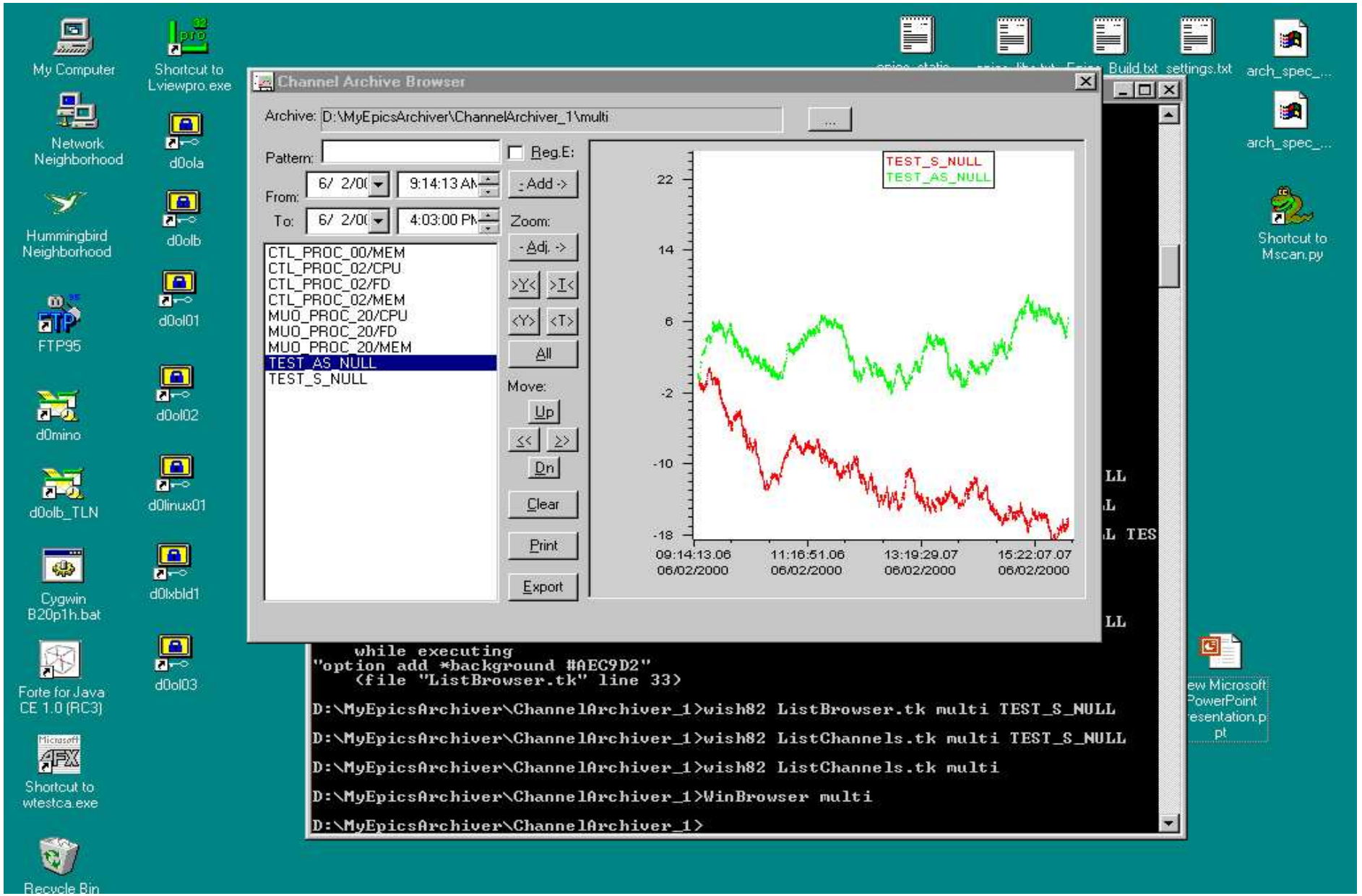
CGIExport Front Page



CGIExport Channel Plot Example



WinBrowser on NT



How to use ChannelArchiver at D0

ChannelArchiver is available both on Linux (online cluster)
and on Windows NT (setup from \\d0server4)

Two ways of using it:

1) Constantly running centralized ChannelArchiver on one of the Linux boxes (at present on d0ol02 or d0ol03)
   in default directory /projects/archive
   Control group will monitor this job and will take care about backing archives to tape every month(?)

2) Private ChannelArchivers
   It's user responsibility to run it and maintain private archives

How to use ChannelArchiver at D0 (cont)

On d0online Linux cluster:

- setup d0online
- setup chan_archiver

Read brief tutorial in $CHAN_ARCHIVER_DIR/doc/AAAREADME

Then:

- prepare configuration file which will contain list of your EPICS variables and control parameters
- start ChannelArchiver as daemon on selected host:

  > start_chan_archiver -c <conf.file> [-d dir-file] [-w work-dir] [-h host] [-p port]

  dir-file: archiver directory file, default dir.{Day-Time}
  work-dir: your working directory, default /projects/archive/
  host: host name where you run your archiver, default: localhost
  port: port number, default 4812
- to stop archiver

  > stop_chan_archiver [-h host] [-p port]

Configuration File

```
Linux:onl->  type beam_loss_mon.cfg

# Group: beam_loss_mon.cfg
# This file was auto-created by the ChannelArchiver Engine
# Instance: EPICS Channel Archiver Engine

!write_period   30      <-  write perion on disk (sec)
!default_period 1       <- scan period (sec)
!get_threshold  20      <- if scan perion is less, monitor!
!file_size      24      <- hours per data file
!ignored_future 6       <- ignore time stamps +6 hours in future
!buffer_reserve 3       <- buffer = write_period/scan_period

SMT_FINGER_IFN045/RLVH  1  Monitor
SMT_FINGER_IFN105/RLVH  1
SMT_FINGER_IFN165/RLVH  1
```

How to use ChannelArchiver at D0 (cont)

Monitoring/managing running archiver:

- using WWW interface: go to http://<host>:<port>
  and you can check the archiver status, disable or add variables or groups of variables, change accessing methods get/monitor, stop running archiver

- using unix tools, like python scripts, ArchiveManager

- Using CGIExport WWW toolkit:
  go to the page http://www-d0ol.fnal.gov/chan_archiver
  open your archive, inspect it's content, plot online any archived variables

How to use ChannelArchiver at D0 on Windows NT

Install ChannelArchiver and appropriate tools on your NT,
Brief instructions are given in file:

\\d0server4\projects\d0epics\ChannelArchiver\AAAREADME

After installation, read AAAREADME file in

\\d0server4\projects\d0epics\ChannelArchiver\current\AAAREADME

Monitoring/managing running archiver: similar to Linux

- using WWW interface on http://<localhost>:<port>
- using command line tools, like python scripts, ArchiveManager
- CGIExport is not set up, use WebBrowser tool