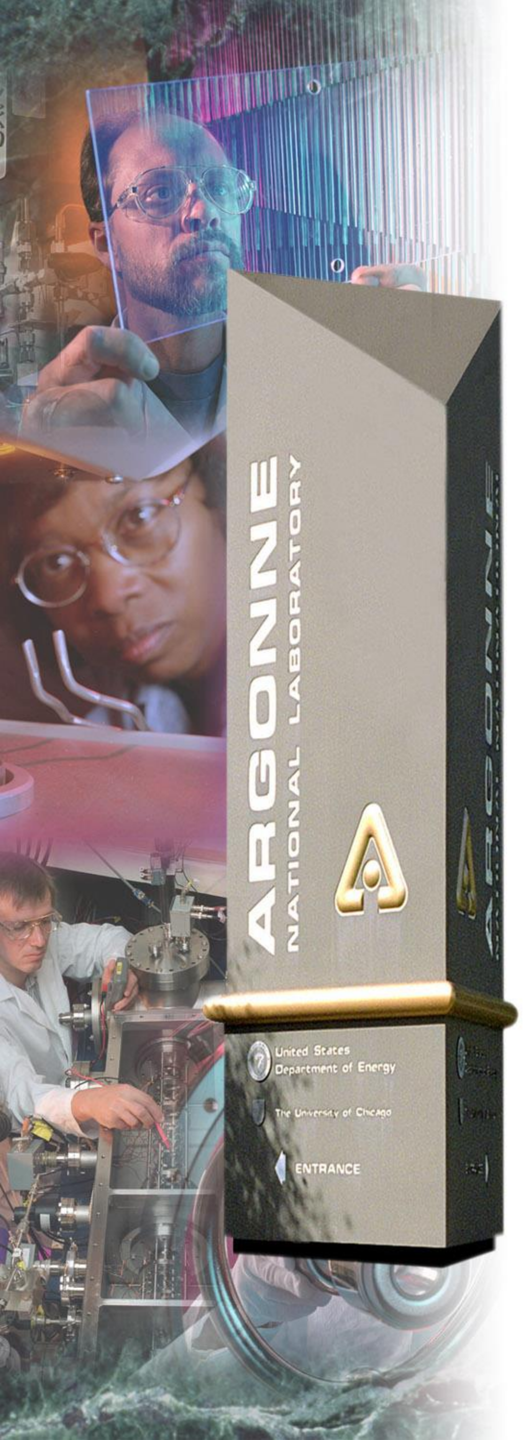


# Selected synApps modules

- **autosave**

  save/restore PV values
- **calc**

  run-time expression evaluation
- **softGlue**

  run-time programmable digital electronics
- **caputRecorder**

  a “macro” recorder for EPICS
- **Demo**

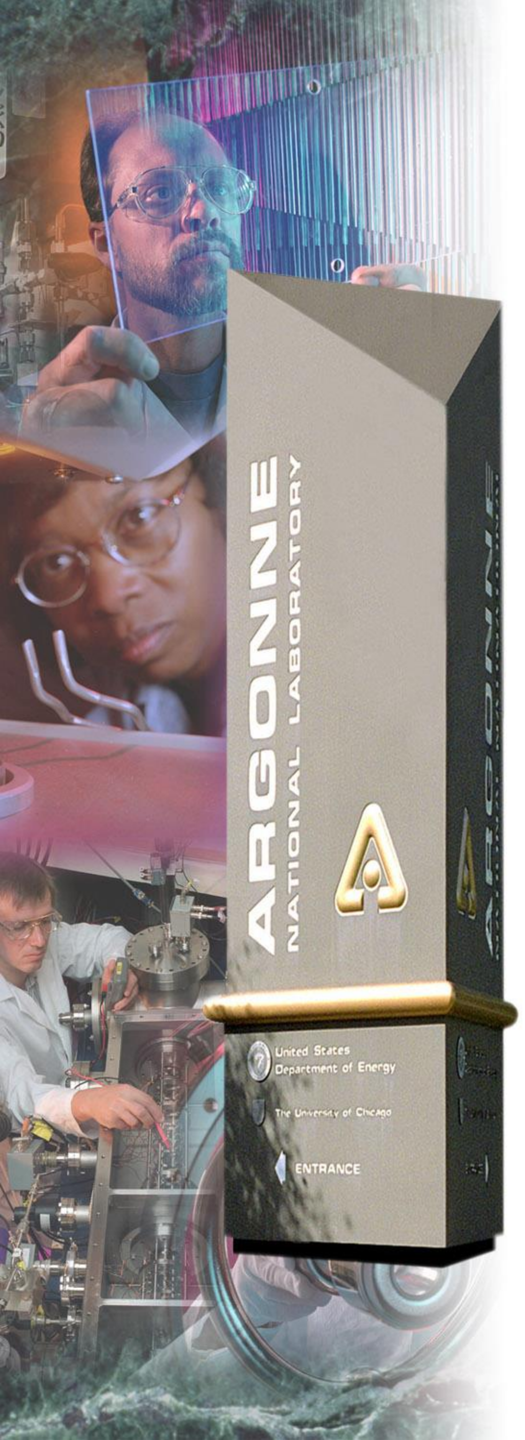
*Tim Mooney*

*3/3/2015*

***Argonne National Laboratory***

Office of Science
U.S. Department of Energy

*A U.S. Department of Energy*
*Office of Science Laboratory*
*Operated by The University of Chicago*



# autosave

*Tim Mooney*

***Argonne National Laboratory***

*A U.S. Department of Energy*
*Office of Science Laboratory*
*Operated by The University of Chicago*

Office of Science
U.S. Department of Energy

## What it does

- **Saves latest values of selected EPICS PVs to a file, and restores those values when the IOC reboots.**
  - Not an archiver; only the latest value is saved
  - When a list list of PV’s is saved, the entire list is written.
  - Each IOC saves and restores its own PVs
- **Can save/restore any scalar or array-valued PV**
  - DBF_MENU, DBF_ENUM PV’s are handled by number.
- **Save operation uses channel access.**
- **Boot-time restore uses static database access.**
- **Manual restore uses channel access.**
- **Arrays are saved and restored with database access.**

# Boot-time restore

- **Three restore options for save files:**
  1) before record/device initialization
     - Motor positions must be restored at this time.
     - Arrays cannot be restored at this time. *
     - PV's that are DBF_NOACCESS before record init (e.g., aSub variable-type fields) cannot be restored at this time. *
  2) after record/device initialization
     - to override record-initialization values
     - Link fields cannot be restored at this time. *
  3) both before and after record/device initialization
     - The 'auto_settings.sav' file is restored at both times.
     - It's not an error to attempt to restore a PV at the wrong time.
     - If you restore a motor position at this time, you override the value read from hardware, without writing to hardware.

* Not illegal, just doesn't work

## Features

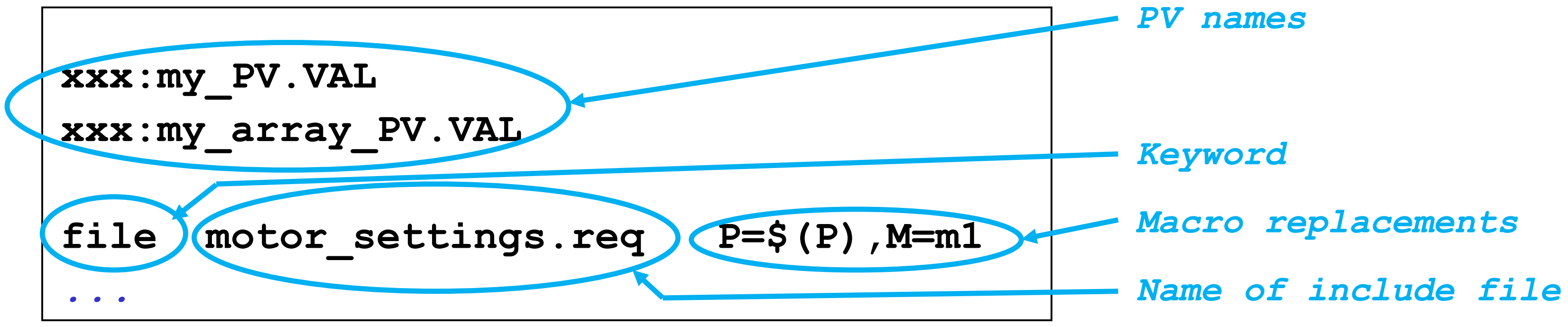
- **PV lists can use include files (e.g., <database_name>.req), include path.**
  - Database developer can supply default include file with database.
  - User can override with custom include file.
- **Save triggers:**

| | |
|---|---|
| - on change of any PV | create_monitor_set() |
| - periodically | create_periodic_set() |
| - on change of a trigger PV | create_triggered_set() |
| - manually | create_manual_set() |

- **User can reload save sets manually. (See also *configMenu*)**
- **Autosave can recover from file-server reboot.**
  - Currently, only on vxWorks and RTEMS
- **User can choose to save recent copies of .sav files.**
- **Autosave reports status via EPICS PV's.**

# File formats

- **Sample request file:**



```
xxx:my_PV.VAL
xxx:my_array_PV.VAL

file motor_settings.req  P=$(P),M=m1
...
```

- **Corresponding save file:**

```
# save/restore V4.4 Automatically generated…
xxx:my_PV.VAL 1.0
xxx:my_array_PV.VAL @array@ { “0” “0.1” … “10.2” }
xxx:m1.DIR 0
xxx:m1.DHLM 100
xxx:m1.DLLM -100
...
<END><xx>          (<xx> means “any character”, usually <\n>)
   OR
<END><xx><xx>
```

# Finding request files

- The command

```
file motor_settings.req P=$(P),M=m1
```

presumes autosave knows how to find `motor_settings.req`

- You tell autosave how with this command:

set_requestfile_path(char *path, char *pathsub)

Examples:

- iocsh:

```
set_requestfile_path($(MOTOR), "motorApp/Db")
```

- vxWorks shell:

```
set_requestfile_path(motor, "motorApp/Db")
```

where the variables MOTOR and motor come from the envPaths and cdCommands files, respectively

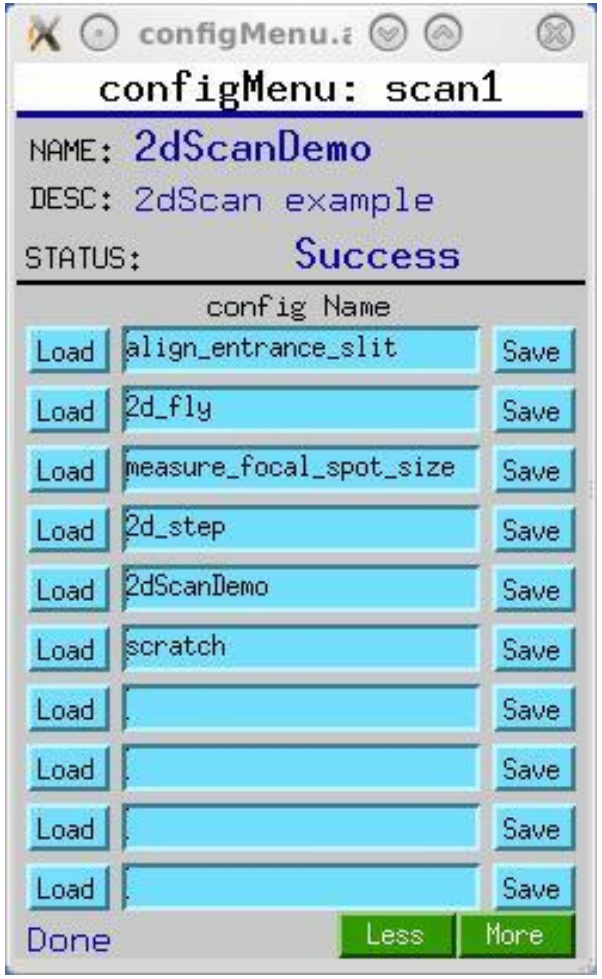
# asVerify (autosave-4-2)

- Client-side verification of a .sav file
- Compares saved values with current (live) values
- Usage:

```
asVerify [-vr] <autosave_file>
        -v (verbose) causes all PV's to be printed out
           Otherwise, only PV's whose values differ are printed.
        -r (restore_file) causes a restore file named
          '<autosave_file>.asVerify' to be written.
```

# configMenu (autosave-5-1)

- **Manual save/restore with an MEDM user interface**
- **Saves/Loads .cfg files (same format as .sav)**
- **Can search directory for, e.g., scan1_*.cfg**
- **Load operation is a manual restore followed by asVerify**
  - Load files can contain any PVs (needn't all contain the same set of PVs)
- **Save operation is directed by a .req file**
  - Save file always contains the PVs in the .req file
  - Can disable save
  - Save writes dated backups
- **Load and Save send callback when done**
  - If written to by ca_put_callback()



NAME: 2dScanDemo
DESC: 2dScan example
STATUS: Success

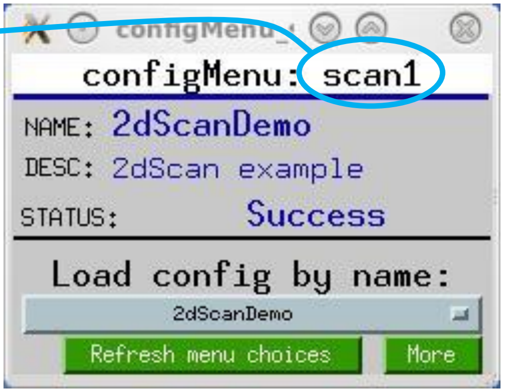
config Name

| | | |
|---|---|---|
| Load | align_entrance_slit | Save |
| Load | 2d_fly | Save |
| Load | measure_focal_spot_size | Save |
| Load | 2d_step | Save |
| Load | 2dScanDemo | Save |
| Load | scratch | Save |
| Load | | Save |
| Load | | Save |
| Load | | Save |
| Load | | Save |

# configMenu

- **Channel-Access clients can use configMenu:**

```
caput("xxx:scan1Menu", "2dScanDemo")
```



configMenu_

configMenu: scan1

NAME: 2dScanDemo

DESC: 2dScan example

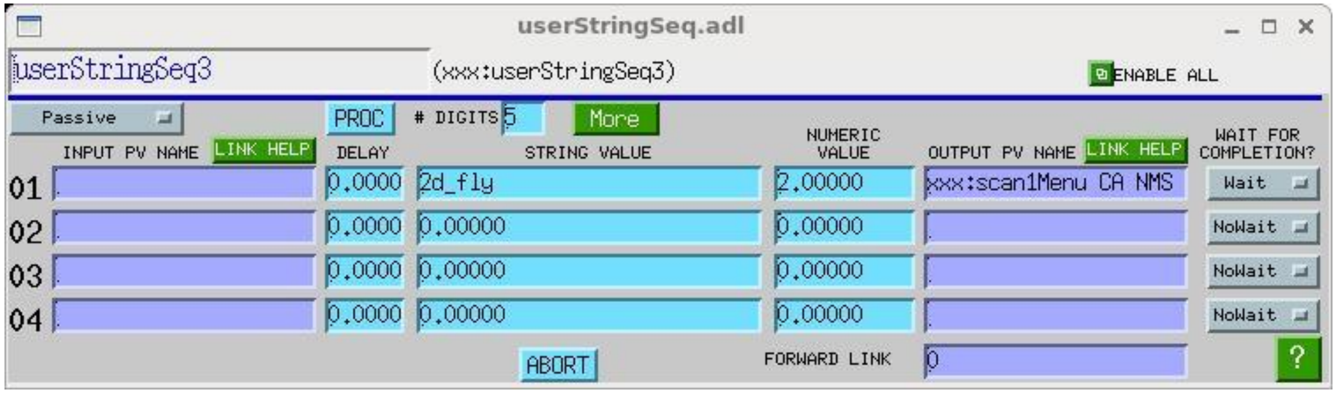
STATUS: Success

Load config by name:

2dScanDemo

```
Python:
epics.caput("xxx:scan1Menu", "2dScanDemo", wait=True)
```

```
String-sequence record:
```



userStringSeq.adl

userStringSeq3 (xxx:userStringSeq3)

# DIGITS 5

| | INPUT PV NAME | DELAY | STRING VALUE | NUMERIC VALUE | OUTPUT PV NAME | WAIT FOR COMPLETION? |
|---|---|---|---|---|---|---|
| 01 | | 0.0000 | 2d_fly | 2.00000 | xxx:scan1Menu CA NMS | Wait |
| 02 | | 0.0000 | 0.00000 | 0.00000 | | NoWait |
| 03 | | 0.0000 | 0.00000 | 0.00000 | | NoWait |
| 04 | | 0.0000 | 0.00000 | 0.00000 | | NoWait |

FORWARD LINK 0

# autosaveBuild (synApps_5_8)

- **Write auto_settings.req, auto_positions.req automatically**
- **Implemented via dbLoadRecords hook**
  - EPICS 3.15, or 3.14.12 with patch
- **When dbLoadRecords is called like this:**

  dbLoadRecords("$(CALC)/calcApp/Db/editSseq.db", "P=xxxL:,Q=ES:")

  **autosaveBuild writes this:**

  file editSseq_settings.req P=xxxL:,Q=ES:
- **Enable with this command:**

  autosaveBuild(“built_settings.req”, “_settings.req”, 1)
- **Use, e.g., by adding this command to auto_settings.req:**

  file built_settings.req P=$(P)
- **Also can add individual PVs to built_settings.req:**

  appendToFile("built_settings.req", “$(P)userStringSeqEnable”)