

Autosave

Kay Kasemir

July 2026

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

Purpose of 'autosave' (aka. 'bumpless save/restore')

Values of some EPICS Records need to persist across IOC reboots

- ✓ 'Set points' entered by user
- ✓ Current 'positions' of motors*

* Fine print:

In some cases, the hardware can do this:

PLC, Parker6k, ...

At IOC startup, Device support 'init()' should then read from hardware.

But what if the hardware lost the settings?

Who's in charge here??

→ Need to think about this case-by-case!

Calibration constants

- Place in *.db or put ``dbpf AbcScaling 3.1428`` into st.cmd?
- Put AbcScaling onto display, then autosave?

How To Use

1. Add 'autosave' module to IOC
2. Configure what to persist

Add Autosave to IOC

1. Define path in `configure/RELEASE` or `../RELEASE.local`:

```
AUTOSAVE=/ics/tools/autosave
```

2. Add binary and DBD info to `xyzApp/src/Makefile`:

```
YourProduct_DBD += asSupport.dbd  
YourProduct_LIBS += autosave
```

3. Use the support module in the IOC `st.cmd` startup file:

```
cd ${AUTOSAVE}  
# Optional database for autosave status  
dbLoadRecords "db/save_restoreStatus.db", "P=$(IOCNAME) :"  
  
# Configure (more later)  
set_requestfile_path("/home/controls/var")  
create_monitor_set(..)
```

Specify What to Save

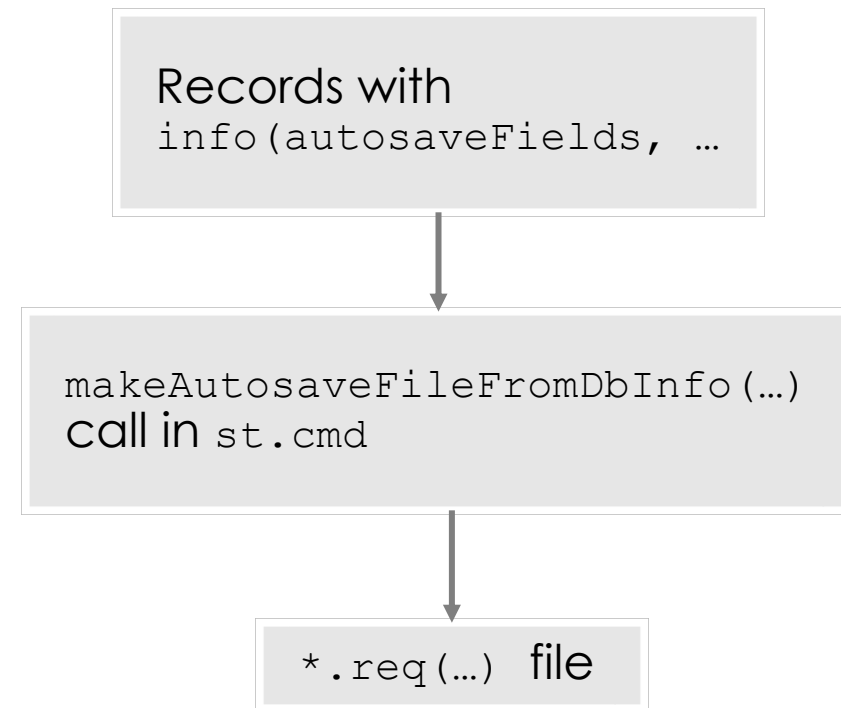
Add annotations to the EPICS database:

```
record(type, "$ (S) :SomeRecord")  
{  
  info(autosaveFields, "VAL LOW")  
  field(LOW, "-10")  
  ..  
}
```

Ultimately, autosave needs a file `somefile.req` that lists what to persist:

```
$(S) :SomeRecord.VAL  
$(S) :SomeRecord.LOW
```

Could create that manually, but suggested approach is to create from EPICS database `info(autosaveFields, ...)` annotations.



IOC Startup: Quite some boilerplate

```
# Configure file locations
epicsEnvSet SAVE_DIR /ics/training/.../$(IOC)

set_requestfile_path("${SAVE_DIR}")
set_savefile_path("${SAVE_DIR}")

# If loading save_restoreStatus.db
save_restoreSet_status_prefix("${IOC}:")
dbLoadRecords "${AUTOSAVE}/db/save_restoreStatus.db", "P=${IOC}:"

# Schedule a maximum of 3 sequenced backups of the .sav file
# every 10 minutes - .sav0, .sav1, .sav2
save_restoreSet_NumSeqFiles(3)
save_restoreSet_SeqPeriodInSeconds(600)

# Arrange for restoring saved values into records
set_pass1_restoreFile("${IOCNAME}.sav")

# Start IOC processes
iocInit

# Create request file and start periodic 'save'
makeAutosaveFileFromDbInfo("${SAVE_DIR}/${IOC}.req", "autosaveFields")

create_monitor_set("${IOCNAME}.req", 30)
```

Autosave

- Some boilerplate for initial setup
 - Later, when adding records, remember `info(autosaveFields, "VAL")`
- IOC now restarts with previously saved values

Autosave Documentation

- <https://epics-modules.github.io/autosave/autosave.html>
- Important nuances Section 8 – “Notes on restore passes”
- Section 10 – Save-file Options