

# Autosave

**Kay Kasemir**

**ORNL/SNS**

**[kasemirk@ornl.gov](mailto:kasemirk@ornl.gov)**

**July 2017**

# 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!

# How To Use

- 1. Add 'autosave' module to IOC**
- 2. Configure what to persist**
  - a) Historically: Designated \*.req list of record names**
  - b) Now: EPICS database annotations create those \*.req files on the fly**

# Add Autosave to IOC

Example: 'Autosave'

## 1. Define path in configure/RELEASE:

```
AUTOSAVE=/home/controls/epics/R3.14.12.2/support/autosave
```

Path to the support directory is usually pulled into a macro, since you often include more than one support module:

```
SUPPORT=/home/controls/epics/R3.14.12.2/support  
AUTOSAVE=$(SUPPORT)/autosave
```

## 2. Add binary and DBD info to xyzApp/Db/Makefile:

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

## 3. Use the support module in the IOC startup file:

```
cd ${AUTOSAVE}  
dbLoadRecords "db/save_restoreStatus.db", "P=demo"  
set_requestfile_path("/home/controls/var"  
create_monitor_set(..)
```

# Specify What to Save

- The preferred way is via annotations in the EPICS database:

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

# IOC Startup

```
# Configure file locations
epicsEnvSet IOCNAME MyIOCName
epicsEnvSet SAVE_DIR /home/controls/var/$(IOCNAME)

save_restoreSet_status_prefix("$(IOCNAME):")
set_requestfile_path("$(SAVE_DIR)")
set_savefile_path("$(SAVE_DIR)")

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("$(SAVE_DIR)/$(IOCNAME).req",
"autosaveFields")

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

# Autosave

- **Some boilerplate for initial setup**
  - **Later, when adding records, remember**  
**`“info (autosaveFields, VAL) ”`**
- IOC now reboots ‘bumplessly’**