



Modifying Hardware Addresses at Runtime

*Andrew Johnson
APS Engineering Support Division*

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*



Outline

- **New feature in EPICS Base R3.14.8**
- **How it's implemented**
 - The `add_record` and `del_record` support routines
- **How to modify existing device support**
- **Some limitations**



New Feature in R3.14.8

- **Permits DTYP and INP/OUT fields to be changed at runtime**
- **Notifies device supports of the change**
 - Changes are only permitted if old and new device layers have registered a Device Support eXtension Table (DSXT)
 - Old device support may cancel a change before it happens
 - New device support may refuse to accept the new record
- **No alterations to record support necessary**
- **Vast majority of device support continues to work as before**
 - Any existing support that looks for changes to the INP/OUT field will need to be converted
 - Without a DSXT, the INP/OUT fields cannot be changed

How it's implemented

- At initialization, device support registers its DSXT with iocCore
devExtend(&myDsxt);
- The DSXT contains function pointers for two routines. From devSup.h:
typedef struct dsxt {
 long (***add_record**)(struct dbCommon *precord);
 long (***del_record**)(struct dbCommon *precord);
} dsxt;
- These routines are optional, use NULL if not provided
static dsxt myDsxt = {myAdd, NULL};
 - This device support is a black hole, no exit is possible

The `add_record` routine

- Called to inform device support of a new record to use
- Actions usually similar to existing `init_record` routine
 - Check INP /OUT field address
 - Initialize DPVT field as necessary for support
 - Set record-specific conversion fields
 - *ai, ao: If LINR=Linear, set ESLO, EOFF*
 - *bi, bo: set MASK*
 - *mbbi, mbbo: set SHFT, MASK*
 - *etc.*
 - Return 0 if OK; non-zero means error
 - *On error, record will be disabled by setting PACT=TRUE*
 - *Set INP /OUT field without error to clear PACT*
- `add_record` is called before `init_record` at startup
 - An `init_record` routine is usually still needed



The del_record routine

- **Called to disconnect device support from a record**
- **Support can refuse by returning non-zero (error)**
 - INP/OUT field will not be changed on error
- **Actions usually only related to device interface**
 - DPVT field will be cleared after an OK return
 - Support must remove all references to this record
- **Records with SCAN=I/O Interrupt will be changed to Passive (calling get_io_int_info) before del_record**
 - Be careful of losing IOS CANPVT pointer values which would cause a memory leak
 - *Old addresses may be re-used later, consider saving the hardware-specific data rather than freeing everything*

How to add to existing support

- Create static DSXT structure
- Register DSXT from pass 0 of DSET's init routine
- Rename old init_record routine to add_record
 - Some record-type specific changes are also necessary
 - *Look at record-type's init_record routine*
- Create new init_record routine
 - Simple record-type specific operations needed here
 - *Usually repeating some code from add_record*
- Add new del_record routine
 - Device-specific operations needed here



Limitations

- User must change DTYP before setting INP/OUT
- Rejection of an address changed via Channel Access will result in a failed ca_put or ca_put_callback
 - some client tools (e.g. seq, caput) do not report failures
- del_record cannot peek at the new INP/OUT address
 - it can see the new DTYP value though
- Ideal for use with ASYN
 - ASYN hasn't been converted to use it yet
- Converted support will no longer compile against earlier versions of EPICS Base
 - Can use **ifdefs** based on epicsVersion.h, but that makes the code harder to read