MakeBaseApp and SNL Exercises

This example uses an instance of the “example” IOC application template, generated using **`makeBaseApp.pl`**

1. Create a <top> directory for your application under your home directory:
   `cd; mkdir example; cd example`
2. Create an example application called “testApp”:
   `makeBaseApp.pl -t example test`
3. Create an IOC directory file for “ioctest”:
   `makeBaseApp.pl -i -t example -p test test`
4. Edit your **`configure/RELEASE`** file and change the SNCSEQ line to this:
   `SNCSEQ=/opt/epics/R3.14.12/modules/soft/seq/seq-2.1.16`
5. Build the application:
   `make`
6. Prepare to execute the application:
   `cd iocBoot/ioctest`
   `chmod +x st.cmd`
7. Edit **`st.cmd`** and remove the **`#`** from the seq line, so that it looks similar to:
   `seq sncExample, user=userHost`
8. Execute the application (finally...)
   `./st.cmd`
   You should now see an **`epics>`** prompt, and be able to use the commands **`dbl`**, **`dbpr`**, etc.
9. Familiarize yourself with the contents of the startup file, the example database files in **`testApp/Db`** and the example sequence program in **`testApp/src/sncExample.stt`**
   Create an EDM screen to display the value of the counting record.
10. Verify proper operation of the sequence program: Run the application and explore the **`seqShow`**, and **`seqChanShow`** commands
11. Add a *stringin* record to the example database and modify the sequence program to update that record's value with the current state name. Display that new record on a CSS screen.
12. Add an error state. If the sequence remains in the high state for more than 5 seconds, it should enter the error state and stay there until the user presses a (new) reset button on the CSS screen. Add the button and additional records as necessary.
13. Verify operation: Rebuild the application and restart the IOC. Add a CSS control for the SCAN field of the example’s saw-tooth record so that you can see whether the error state is entered if you slow down the counter.