

MPS Config

- As of June 17th 2011, there may be a bug in reloading the Config DB after faults were removed.

1. [#Setup](#)
 - a. [#Environment](#)
 - b. [#Dependencies](#)
 - c. [#Test inside Eclipse](#)
2. [#Development](#)
 - a. [#Overview](#)
 - b. [#Classes and Functions](#)
3. [#Release with Eclipse](#)

Setup

Environment

- Check out the CVS module physics/mps/mpsconfig into your workspace
 - From now on, the root of project is referred to as \$MPSCONFIG_ROOT
- Use ant to build the mpsconfig.jar

Dependencies

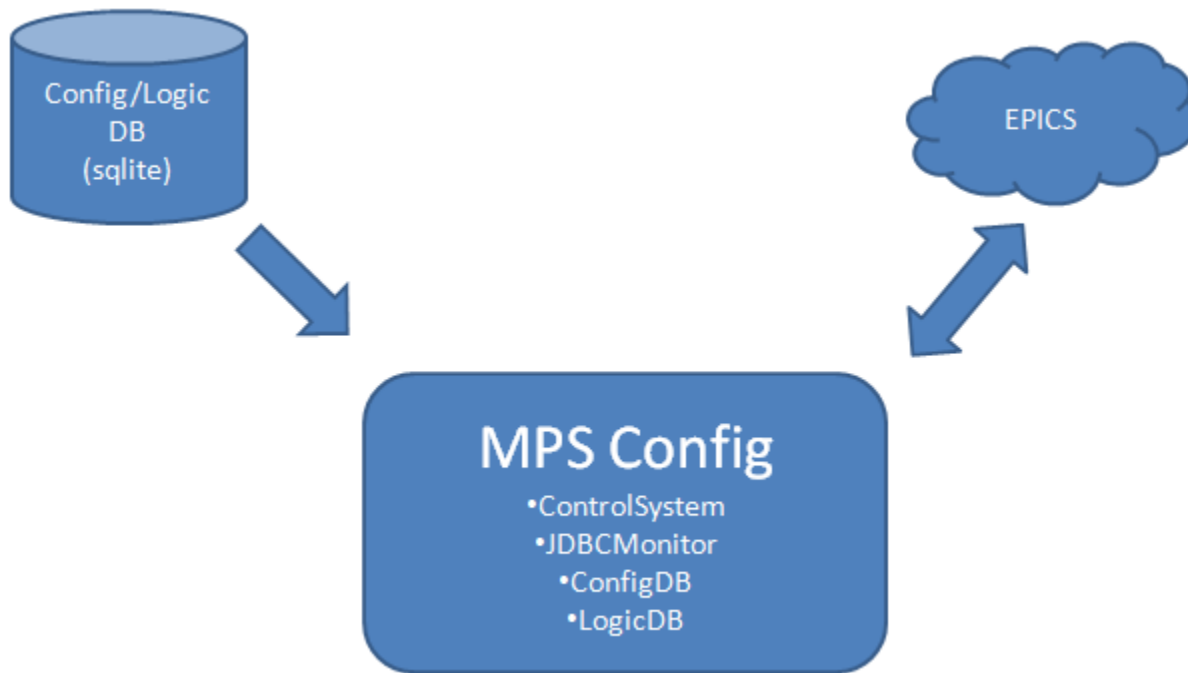
- ezJCA (version R0-0-10)
- javainterfaces (current version)
- Oracle JDBC driver (under \$MPSCONFIG_ROOT/lib/)
- Sqlite JDBC driver (under \$MPSCONFIG_ROOT/lib/)
- xal4lcls
 - ext.jar (current version)
 - xal.jar (current version)

Test

- To test your changes, you can run the [MPS GUI](#)
 - Check out the MPSGui project
 - Add the MPSConfig project to MPSGui's build path
 - Remove the library mpsconfig.jar from MPSGui's build path and replace it with the one you have built.

Development

Overview



Classes and Functions

`edu.stanford.slac.mps.epics.ControlSystem`

- Monitors current Macro state numbers from IOC:BSY0:MP01:TTBLST.VALA
- Bypasses faults to a value for a time duration
- Monitors bypassed faults and end times
- Given a fault name, gets current state, type, and timestamp of the fault
- Given a PV, gets beam rate names (MPS and actual)
- Translates between EPICS and Java time

`edu.stanford.slac.mps.jdbc.JDBCMonitor`

- Loads all data from Config/Logic DB
 - Config DB URL prefix: `jdbc:sqlite:/usr/local/lcls/epics/iocTop/MachineProtection/mpsConfiguration/database/`
 - Logic DB URL prefix: `jdbc:sqlite:/usr/local/lcls/epics/iocTop/MachineProtection/mpsConfiguration/algorithm/`
- Checks periodically for new versions
 - Current Config DB version is stored in IOC:BSY0:MP01:DBVERS
 - Current Logic DB version is stored in IOC:BSY0:MP01:ALGRNAME

`edu.stanford.slac.mps.jdbc.config.ConfigDB`

- Singleton
- Serves as a facade to information about faults
- SQL queries are in the file `sqlite.properties` in the package `edu.stanford.slac.mps.jdbc.config`

`edu.stanford.slac.mps.jdbc.logic.LogicDB`

- Singleton
- Serves as a facade to information about Macros (and states), ignore conditions, current MacroStates
- SQL queries are in the file `sqlite.properties` in the package `edu.stanford.slac.mps.jdbc.logic`

Release using cram

- Add a note in `$MPSGUI_ROOT/RELEASE_NOTES`
 - Increment the tag version accordingly
- Commit to CVS
- Tag with `mpsconfig-R#-#-#`
- Check out the tagged version into your workspace on dev using `eco`

```
$ cd ~/workspace
$ eco
Enter name of module/package to checkout: mpsconfig
Enter name of tag or [RETURN] to use HEAD>mpsconfig-R0-0-11
Using mpsconfig-R0-0-11. The name of the directory will be mpsconfig-R0-0-11.
cvs checkout -P -r mpsconfig-R0-0-11 -d mpsconfig-R0-0-11 mpsconfig
cvs checkout: Updating mpsconfig-R0-0-11
...
$ cd mpsconfig/mpsconfig-R0-0-11/
$ ant
```

- Ant should generate the jar file in mpsconfig.jar
- You can then use cram push and cram upgrade to upgrade the mpsconfig package
- Note that mpsconfig is a dependent jar and is used by the other MPS packages. You will need to build new versions of these packages to use the new version of mpsconfig by updating their jars using



Unknown macro: 'html'