MX Data Acquisition & Control Library

The MX library has several goals:

- Designed to be a uniform and portable interface to motors and other devices.
- Constructed as a library so that it may be embedded in other applications.
- Intended to achieve some degree of platform independence for data acquisition code.
- Intended to allow for rapid development of new drivers (2-3 days in many cases).
- Also may be used to extend the capabilities of control systems such as EPICS.
- MX is used both in client applications and in servers.

A Uniform Interface

- EPICS GUI screens for motors generally look fairly similar to each other.
- The programming interfaces for EPICS motors are not standardized very much.
- MX provides a uniform programming interface for common operations.
- MX has been used to control both OMS VME 58 stepping motors and Delta Tau PMAC servo motors.
- Can also be used to control non-EPICS based motors, so that the same application may be used in your lab, at NSLS, and so forth.

MX Motor Driver Functions

- motor_is_busy
- move_absolute
- get_position
- set_position
- soft_abort
- immediate_abort
- positive_limit_hit
- negative_limit_hit
- find_home_position

plus several functions for setting up motors and saving motor parameters.

MX Records

The MX library is based on a linked list of structures called MX_RECORDs. MX_RECORDs are classified by *superclass*, *class*, and *type*. The currently existing types are:

Interfaces - RS-232, GPIB (*soon*), CAMAC, and GENERIC.

Devices - Analog Input, Analog Output, Digital Input, Digital Output, Motor, Encoder, Scaler, and Timer.

Scans

Linear - *Input, Motor,* $\theta - 2\theta$*, and Slit*

List of Positions - *File*

XAFS

Segmented - *Region*

Variables

Inline - *string, char, uchar, short, ushort, int, uint, long, ulong, float, double*

Network Servers

MX

MX Servers

Using Channel Access

- Provides another mechanism for interfacing devices to EPICS.

- Allows the use of vendor distributed drivers by running them in the environment they expect (Windows 95, Windows NT, etc.)

Using Other Protocols

- Allow MX to provide services to clients written for another server.

- Allow MX to be run in multi-process mode in a home lab.

MX Network Access

MX uses a field based scheme for network communications that is similar to EPICS.

Examples of motor fields:

- setpoint
- position
- backlash_correction
- negative_limit
- positive_limit
- raw and dial versions of the above.
- scale
- offset
- units
- busy
- soft_abort
- immediate_abort
- negative_limit_hit
- positive_limit_hit
- home_search
- set_mode

MX Servers For Channel Access

Using Only CA Client Functions

A set of soft records in an EPICS IOC are used as front end mailboxes for the MX server. The process is as follows:

1. A CA client writes to one of the soft records.
2. The MX server uses CA client monitors to be notified when one or more of the mailbox records is written to.
3. The value written is used as a command to perform some operation. The MX server performs the operation and the results are written back to other soft records in the IOC.
4. The IOC forwards the results on to the CA client, if it has appropriate monitors set up.

Currently, this is implemented using a set of soft EPICS records. Ideally, this would be done using a soft XFD motor record for compatibility with *spec*.

MX Servers For Channel Access

Using Portable Channel Access Server

- Probably will have better performance than the first server, but will take longer to develop.

- Likely to be preferable in the long term.

Applications

Current

- “motor” - command line based client program.
- MX server - can be contacted via EPICS and native MX protocol.

Later

- GUI-based spectroscopy program.
- Bruker GGCS emulator (diffractometer controller)
- Connect an MX server to applications using other protocols.
- Use MX with portable channel access server.

The “motor” Program

Much of the scan, datafile, etc. functionality of the “motor” program comes from the underlying MX library:

- Simple command line based program that can run motors and data acquisition scans.
- Performs scans using the scan types that are built into the MX library itself.
- Can handle multiple datafile types (three are currently defined). Would like to use the NEXUS format when it is ready.
- Currently plots scans using Gnuplot, but the hooks for other plot types are there.
- Intended primarily for beamline staff, but has been used for a number of experiments including XAFS.

GUIs for MX

- MEDM - Uses screens derived from the VME58 motor record screen.

- Tcl/Tk and IDL - Useful since both are available on Unix, Microsoft Windows, Macintosh and VMS.