Epics Applications Documentation

Release 1.0

Matthew Newville

September 14, 2011

CONTENTS

PyEpicsApps contains several Epics Applications written in Python, using the pyepics module. Many of these are GUI Application for interacting with Epics devices through Channel Accesss. The programs described here are mostly meant either as end-user applications, or as examples of end-user applications.

CHAPTER

ONE

OVERVIEW

An *Epics Instrument* is simply a grouping of low-level parameters (Process Variables) as exposed through Epics Channel Access. At first, this may not seem very interesting. However, Epics Instruments allows you to

The Epics Instruments application allows you to group these components into a logical group – an Instrument. Once an Instrument has been defined, you can then save and restore settings for this Instrument. Furthermore, these settings are automatcally saved in a single, portable file for later use.

Epics Channel Access gives a simple and robust interface to its lowest common unit – the Process Variable or PV. The Epics control system also provides sophisticated ways to express and manipulate complex devices, both physical and virtual. Creating such devices and defining their behavior is generally done by well-trained programmers. The application here uses a much simpler approach that can expose some categories of "Settings" that may need to changed en masse, and returned to at a later time.

As defined here, An Epics Instrument is simply a named collection of Epics Process Variables (PVs). The PVs do not need to be physically related to one another nor be associated with a single Epics Record or Device. Rather, an Instrument is defined at the level of the Epics Channel Access client, allowing a station scientist or engineer to use their own grouping of PVs as an abstract "Instrument". A simple example would be a pair of motors that work together to move some device.

In addition to a name and a set of PVs, an Instrument has a set of "named Positions". At any point, the current values of an Instruments PVs can be saved as its Position. And, of course, the named Positions can then be restored simply by selecting the Position.

CHAPTER

TWO

DOWNLOADING AND INSTALLATION

Epics Instruments is a GUI Application. For Windows, you can download and install from a binary installer.

For Linux and Mac OS X, building and installing from source are currently the principle options (a Mac OS X App will be available soon).

2.1 Downloads

The latest installer

Download Option	Location
Windows Installer	EpicsInstrumentInstaller.exe
Source Kit	coming soon

CHAPTER

THREE

USING EPICS INSTRUMENTS

3.1 Getting Started

To run Epics Instruments, click on the icon. You will see a small window to select an Epics Instrument File. If this is your first time using the application, choose a name, and hit return to start a new Instrument File. The next time you run Epics Instruments, it should remember which files you've recently used, and present you with a drop-down list of Instrument Files.

3.2 Defining an Instrument

3.3 The Instrument File

All the information for definitions of your Instruments and their Positions are saved in a single file – the Instruments file, with a default extension of '.ein' (Epics INstruments). You can use many different Instrument Files for different domains of use.

The Instrument File is an SQLite database file, and can be browsed and manipulated with external tools. Of course, this can be a very efficient way of corrupting the data, so do this with caution. A further note of caution is to avoid having a single Instrument file open by multiple applications – this can also cause corruption. The Instrument files can be moved around and copied without problems.

CHAPTER

FOUR

IMPLEMENTATION NOTES

Epics Instruments is built with Python, PyEpics, wxPython, and SQLite.