

PVSSProxy

The first piece of the MACS procedure framework (ProShell)

Angela Brett

Procedure Framework

A framework which can run procedures that perform operations on devices and sets of devices (working sets.)

A procedure is a C# class implementing a certain interface, which will be loaded dynamically from a file.

Procedure Framework

The procedure framework will communicate with
devices
using PVSS

PVSS

PVSS

PVSS is a SCADA (Supervisory Control and Data Acquisition) system. It can be used to connect to hardware devices to control them and acquire data from them.

PVSS has a scripting language and a graphical user interface, but a procedure framework implemented using these would run too slowly.

The PVSS API allows users to write their own programs in C++ which can access the data and control the hardware devices via PVSS datapoints.

Procedure Framework

**How do we bridge
this gap?**

PVSS

Procedure Framework

SWIG

PVSS

SWIG

SWIG (Simplified Wrapper and Interface Generator) is free software that can create wrappers that allow C or C++ APIs to be used from other languages.

I used SWIG to create a wrapper for the PVSS API. It took about a month to fine tune the wrapper and write prototype code to ensure that all the features of the PVSS API could be used.

Procedure Framework

PVSSProxy

SWIG

PVSS

PVSSProxy

PVSSProxy is a library which allows easy use of the SWIG wrapper without knowledge of the underlying API.

All of the main functionality of PVSS is made available using common C# patterns:

- Asynchronous calls use C# delegates.
- Datapoints values can be set and retrieved using native C# data types.
- Errors are raised as exceptions.
- PVSSProxy manages the PVSS event loop, so client code only needs to call one method to connect to PVSS.

PVSSProxy

PVSSProxy should also make some things easier to code than they would be in a PVSS script.

DatapointName class simplifies the use of datapoints. Instead of remembering how to make a full datapoint name string such as:

```
Datapoint.element:_config.Detail._attribute
```

You create a DatapointName object with only the essential properties, and the rest can be filled in automatically depending on what you are doing.

DatapointName will also allow wildcards to get names and values of many datapoint elements at once.

PVSSProxy

To do:

- Add more source code documentation
 - Add the ability to disconnect from PVSS
 - Make timeouts configurable
 - Add logging using log4net
 - Allow wildcards in datapoint names
 - Add more unit testing
- ...and implement the procedure framework.

Procedure Framework

PVSSProxy

SWIG

PVSS