Data Provider Cookbook

PERL for SOS 1.0 (ASCII)

Supplemental Notes for MS Windows Install
Version 1.0

Robert Leeman
SmartBay
St. John's, NL, Canada
September 14, 2009

1. Preface

These notes were written following the recent Perl-based SOS (ASCII) implementation for *SmartBay* buoys, located in Placentia Bay, Newfoundland and Labrador, Canada (http://www.SmartBay.ca). This information is meant to be used in conjunction with the primary implementation notes as provided by **OOSTethhys** (i.e. *Data Provider Cookbook – Perl for SOS 1.0 (ASCII and RDMS Sources*)), and should be considered supplemental information for Windows users with the specific environments described below (or very similar). This document focuses primarily on installing and configuring ActivePerl and the required ActivePerl modules. Please note that the author had never used Perl prior to this SOS installation so when I am stating the obvious for veteran Perl users, I apologize in advance. Also, I do not warrant this information so use with that understanding and please forgive any errors or omissions.

2. Introduction

The implementation was done on two web server computers: a localhost Development/Test Server and our *SmartBay* Production Server.

Our Environments:

Development/Test Server
MS Windows XP Professional, Version 2002, SP 2
Intel Pentium 4
Webserver: Apache 2.2.4
Database: MySQL Server 5.0
Perl 5.10.0 build 1005 by Active State (ActivePerl from www.ActiveState.com)

PHP 5.2.5 is also installed on this server, although not used for the SOS.

Production Server
MS Windows Server 2003, Standard Edition, SP 2
Intel Xeon
Webserver: Apache 2.2.4
Database: MySQL Server 5.0
Perl 5.8.8 build 817 by Active State (ActivePerl from www.ActiveState.com)

PHP 5.2.4 is also installed on this server, although not used for the SOS.

3. Perl Install

Prior to this project Perl was not installed on our development server. The current version of ActivePerl for Windows (x86) (5.10.0) was downloaded and installed from www.activestate.com/activeperl/. We experienced no difficulties with that.

ActivePerl 5.8.8 was already installed on our production server.

As usual, you should read any Release, Readme, FAQ notes, etc. regarding your install. These are installed in C:\Perl\html\...

A note about paths: Our Perl 5.8.8 install uses the path: C:\usr\… but otherwise seems similar to 5.10.0. In these notes "Perl" and "usr" in the path may be interchanged, and depends on the system's specific Perl location.

4. Configuring Apache for Perl

The Perl document file: **C:\Perl\html\faq\Windows\ActivePerl-Winfaq6.html** contains information for configuring various webservers for Perl support. For our particular environment it includes a section called **"How do I configure Apache 2.0.x to support ActivePerl?"**. These instructions were implemented and proved to be simple and accurate. Since this information is readily available in the cited *faq* file it is not repeated here.

I ran a small test **cgi** script after to verify the Apache configuration. For novice Perl users there is a section in the *faq* document above that tells you how to do this.

Definitely verify that Apache is properly handling Perl cgi scripts before continuing.

5. Installing Perl Modules

Several Perl modules must be installed to support the SOS implementation. Two modules are specifically cited in the "Perl Cookbook" but I actually needed to install others, as well. My full list was as follows:

XML-LibXML-Common
XML-SAX
XML-NamespaceSupport (this installed automatically with XML-SAX with 5.8.8)
XML-LibXML
XML-LibXML-XpathContext (this installed automatically with XML-LibXML with 5.8.8)

The install order is somewhat important because of some module dependencies so I recommend installing in the order shown above.

Very Important: **libxml2.dll**

You must have **libxml2.dll** on your system before installing these modules. And it must be the correct version for your ActivePerl version.

This file must be placed in C:\Windows\System32

There are various versions of **libxml2.dll** available and it was not readily apparent to me how to determine which file version goes with a particular version of Active Perl. With some minor trial and error, I got a version that worked with 5.10.0 at:

http://theoryx5/uwinnipeg.ca/ppmpackages/scripts/

libxml2.dll 04-Dec-2006 23:14 906K

With this file placed in the ..\System32 directory I then installed the modules (see procedure below).

With 5.8.8 I took a different approach. When I installed the module XML-LibXML-Common (see procedure below) the install process offered to fetch the required **libxml2.dll.** I allowed this and the file was placed in C:\usr\bin. The module reported as successfully installed. I then copied this **libxml2.dll** to C:\Windows\System32 and continued to install the rest of the modules.

Installing Modules

There are specific module versions for 5.10.0 versus 5.8.8 so it's important to use the right versions.

With 5.10.0, I downloaded the module install packages from:

http://cpan.uwinnipeg.ca/PPMPackages/10xx/

And then installed them from a Command Window prompt, in the directory where the .ppd files were downloaded, using this typical command:

```
prompt>ppm install XML-LibXML.ppd
```

With 5.8.8, I did the following typical procedure to install modules:

- Open a Command Window
- Start **PPM** (Programmer's Package Manager):
  ```
  prompt>ppm
  ```
- Run module install command:
  ```
  ppm>install http://theoryx5.unwinnipeg.ca/ppms/XML-SAX.ppd
  ```

The 5.8.8 approach was a little easier and managed the versioning automatically.

5.8.8 Module Installation Comments

With the 5.8.8 XML-SAX install I was asked "Do you want alter ParserDetails.ini [yes]" to which I answered default "yes" and the install proceeded correctly.

With the XML-LibXML module install I got the same question to which I again answered default "yes". This caused an error and the module install failed. I then repeated the install and answered "no" to this question. The installation was then successful.

About PPM

PPM, the Programmer's Package Manager or Perl Package Manager, was installed automatically with our versions of ActivePerl. With 5.8.8, it seems to be strictly a command line utility whereas with 5.10.0 it provides a handy Windows interface. In either case you can start it by type **ppm** at a Command Window prompt.

How Can You Tell if Modules are Installed?

Open a Command Window and start **ppm**.

With 5.10.0, the PPM window automatically lists all installed modules.

With 5.8.8, you can use a typical command like this:

```
ppm>query XML
```

to list all installed modules with "XML" in their names.

Or, in either version you can type a typical command like this at a command prompt:

```
prompt>ppm query XML
```

How Can You Test **XML-LibXML** and **libxml2.dll** ?

At a Command Window prompt type:

```
prompt>perl -MXML::LibXML::Common -e "print 1"
```

If **libxml2.dll** is correct and available this command will run. Otherwise, it will report errors.

6. Configuring Perl for MySQL Database

Two modules had to be installed to use our MySQL database (where we store our buoy observations), as follows:

prompt>ppm install DBI (the Perl Database Interface)
prompt>ppm install DBD::mysql (for version 5.10.0, the MySQL DB driver)

With 5.8.8, I installed the MySQL driver like this:
prompt>ppm install http://theory5x.uwinnipeg.ca/ppms/DBD-mysql.ppd

7. Implementing SOS – ASCII Version

Once all Perl modules were installed and working correctly the remaining SOS install process followed as written in the "SOS Server CGI Cookbook".

The only minor difference for Windows being that the first line in **oostethys_sos.cgi** should be written as:

```
#!c:\perl\bin\perl.exe
```

The primary difference is back-slashes for Windows versus forward slashes for other environments (and "perl" is "usr" on our 5.8.8 system).

8. CF Metadata

For configuring our **sos_config.xml** file and writing the Perl scripts to extract our ASCII observation data files from our MySQL database I found the **CF Metadata** website to be very useful.

> http://cf-pcmdi.llnl.gov/documents/cf-standard-names/standard-name-table/12/cf-standard-name-table.html

I have written Perl scripts to automatically create the required observation data files several times per hour, in sync with the update cycle on our buoys.

9. Acknowledgement

I would like to thank Eric Bridger at GoMOOS for his invaluable advice and guidance during our implementation.

Robert Leeman
SmartBay
September 14, 2009