UPS Tools
Sample Code Documentation

Version: 2.00

NOTICE

The use, disclosure, reproduction, modification, transfer, or transmittal of this work for any purpose in any form or by any means without the written permission of United Parcel Service is strictly prohibited.

© 2010 United Parcel Service of America, Inc. All Rights Reserved.

Table of Contents

1. Introduction

IMPORTANT: In order to successfully test transactions, the Client will need to submit transactions containing their own:

- *Access License*
- *UserID*
- *Password*
- *Shipper Account*
- *1Z numbers*

UPS tools' sample code is developed to assist developers in quick development of UPS API applications. Developers may use sample code as reference for how to create, populate various UPS Tools/API requests, and invoke them. UPS samples are available in JAXB 2.1, PERL and PHP technology.

2. JAXB 2.1 Sample Code Naming Convention

Sample Java code classes use following naming convention:
Technology Name + UPSToolName +Client

Example 1: **JaxbShipConfirmClient.java**
Jaxb – Technology name
Ship – UPS Tool Name
Client – Word 'Client' is attached to indicate it is a client program to invoke UPS tool/API.

Example 2: **JaxbVoidClient.java**
Jaxb – Technology name
Void – UPS Tool Name
Client – Word 'Client' is attached to indicate it is a client program to invoke UPS tool/API

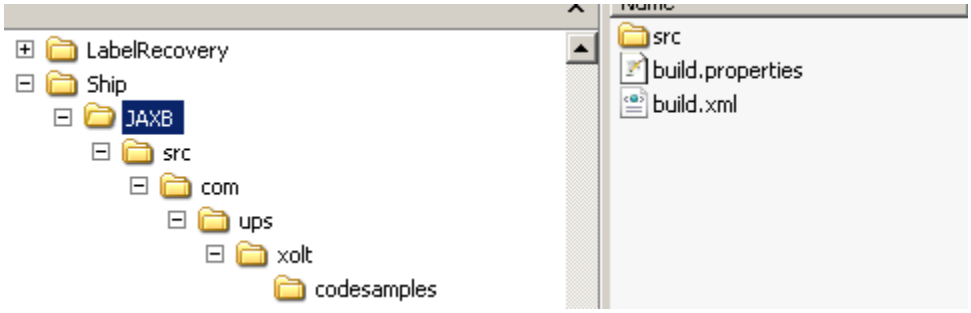
3. JAXB 2.1 UPS Tool Sample Code using ANT build

- UPS Tool JAXB sample codes are provided with build.xml. Sample codes have the following directory structure:

- Root directory (name of root directory web service name)
    - src directory: contains JAXB sample client java code

- build.xml: ant build script to clean, generate stubs, compile and execute sample code
- build.properties: is used by build.xml and client class for reading configurable properties)

Below screenshot shows the directory structure of Ship Tool Java sample code:



Steps for executing build.xml

In order to run build.xml, a developer needs to have Apache Ant installed and configured on the same machine. Apache Ant is a Java based build tool. For more details, please refer to Apache Ant web site at http://ant.apache.org/.
The response and response status can be found in XOLTResult.xml file. Please refer build.properties for the location of XOLTResult.xml file. The default build target is to run the complete sample program. But for your own purpose, you may select the specific target in the build.xml to run.

Below screen shot shows the contains of the build.properties

```
url= Ship Tool URL
accesskey=your access Key
username=your user name
password=password
build=build
src=src
build.classes=build/classes
gen-src=gen-src
dist=dist
conf=etc
jaxb.home=JAXB2.1 Root
requestxsd=Tool Request schema
responsexsd=Tool Response schema
accessrequestxsd=Tool Access Request schema
requestpackage=com.ups.xolt.codesamples.request.jaxb
responsepackage=com.ups.xolt.codesamples.response.jaxb
accessrequestpackage=com.ups.xolt.codesamples.accessrequest.jaxb
clientclass=com.ups.xolt.codesamples.JAXBTrackClient
out_file_location=XOLTResult.xml
```

Below is the sample response you will get when you open the XOLTResult.xml File after excuting the client program.

```
- <ShipmentConfirmResponse>
  - <Response>
    - <TransactionReference>
        <CustomerContext>JAXB Test Client</CustomerContext>
      </TransactionReference>
      <ResponseStatusCode>0</ResponseStatusCode>
      <ResponseStatusDescription>Failure</ResponseStatusDescription>
    - <Error>
        <ErrorSeverity>Hard</ErrorSeverity>
        <ErrorCode>250003</ErrorCode>
        <ErrorDescription>Invalid Access License number</ErrorDescription>
      </Error>
    </Response>
  </ShipmentConfirmResponse>
```

4. Build UPS Tool Sample Code in Perl

- Requirements
- Naming Convention
- Windows Development
- UNIX/Linux Development

Requirements

1. Perl 5.8 or above
2. Perl Package Manager
3. XML::Compile version 1.22 or above
4. XML::LibXML::Simple version 0.91 or above
5. LWP::UserAgent version 6.03 or above
6. HTTP::Request version version 6.02 or above
7. Data::Dumper version 2.131 or above

Naming Convention

Sample Perl code use following naming convention:
Technology Name + UPSToolName +Client

Example 1: **XMLCompileShipConfirmClient.pl**
XMLCompile – Technology name
Ship – UPS Tool Name
Client – Word 'Client' is attached to indicate it is a client program to invoke UPS tool/API.

Example 2: **XMLCompileVoidClient.pl**
XMLCompile – Technology name
Void – UPS Tool Name
Client – Word 'Client' is attached to indicate it is a client program to invoke UPS tool/API

- UPS Tool Perl sample code is provided. Perl sample code have the following directory structure:

- Root directory (name of root directory)
  - Perl directory: contains Perl sample client code

Windows Development

4.1 Installing Perl

ActivePerl distribution should be used for Perl development. To download msi executable please refer to the Reference section at the end of this document. A link will be provided to download site.

4.2. Installing Perl modules

We recommend using Perl Package Manager that comes included with ActivePerl distribution. The Perl Package Manager (PPM) can be used to download and install Perl modules automatically.

4.2.1 Firewall/Proxy Configuration

Using PPM requires Internet access to Perl repository sites to download Perl packages. Below in red is the error message that occurs when PPM cannot connect to Internet:

Downloading ActiveState Package Repository packlist ... failed 407 Proxy Authentication Required

To prevent this error message you must define a SYSTEM variable inside Environment Variables. Under Environment Variables, create a system variable name "http_proxy". Once the system variable and value has been defined restart computer.

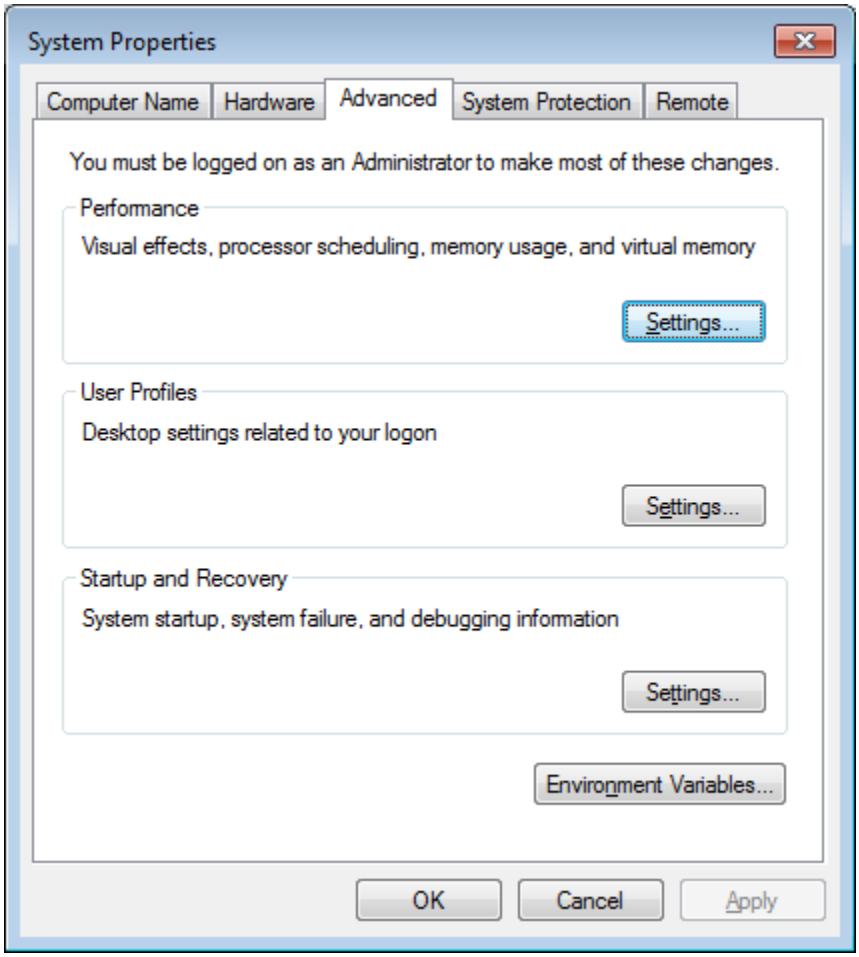


Figure 1: Environment Variables under System Properties

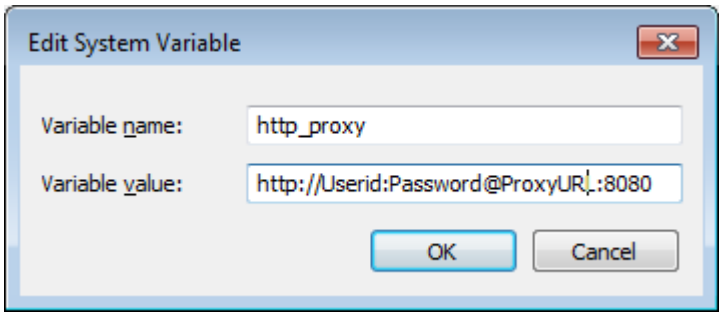


Figure 2: Userid and Password should be the network id and password used to log onto your network or computer. ProxyURL should be the hostname used to connect to Internet. Please consult your IT administrator to get proxy details.

4.2.2 Perl Package Manager (PPM)

Adding a repository

To add repository, open a command prompt and type the following command

ppm repo add trouchelle.com http://trouchelle.com/ppm10/

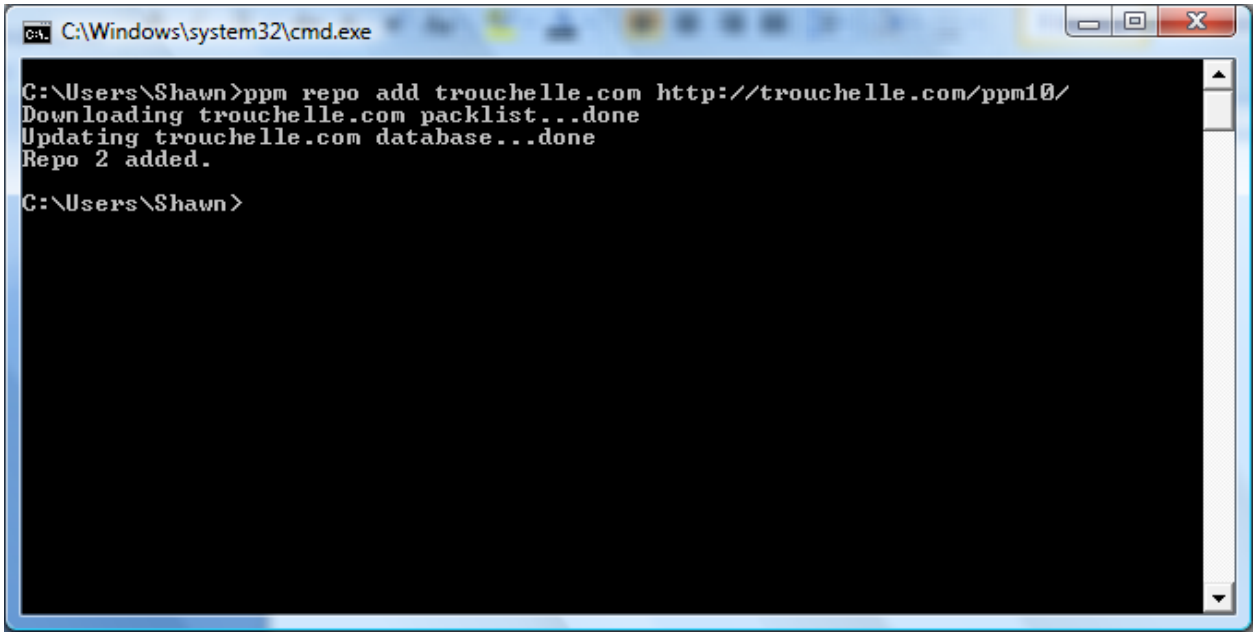


Figure 3: Creating a repository

Installing a Perl module

To install a Perl module, open a command prompt and execute the following commands in chronological order.

Search command

ppm search XML::Compile

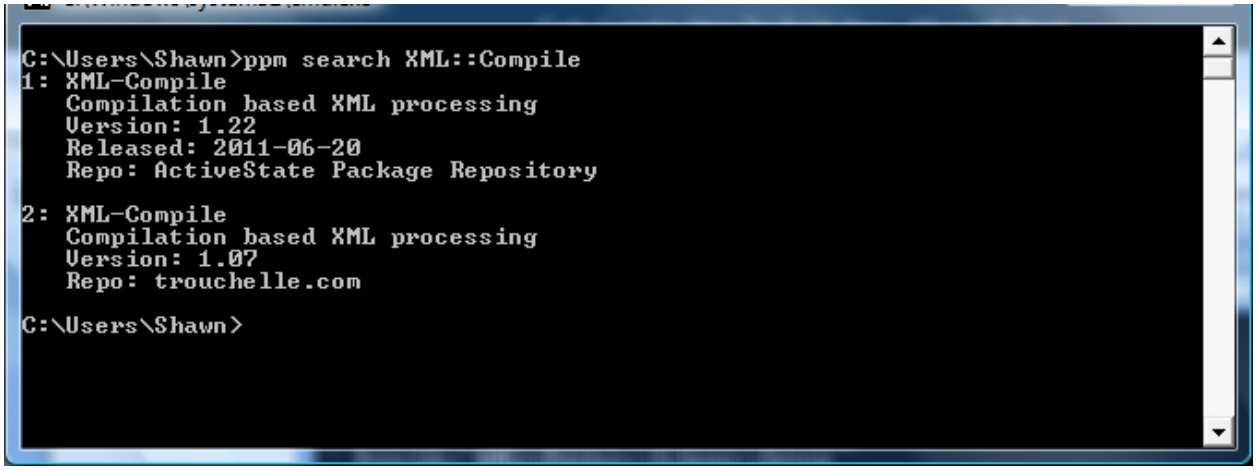


Figure 4: Module search for XMLCompile

Install command

ppm install 1

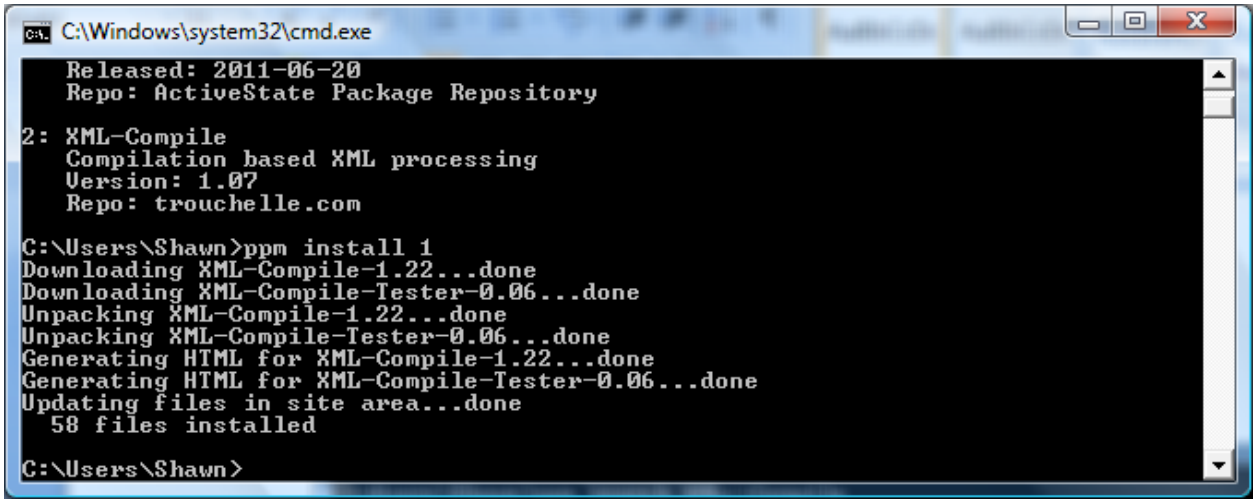


Figure 5: Install XMLCompile using ActiveState Package Repository site.

We could have chosen to use trouchelle.com repository but it has an older version of XMLCompile. To download XMLCompile from trouchelle.com repository, we could have entered "**ppm install 2**" instead of "**ppm install 1**".

4.2.3 Module dependencies

PPM handles the module dependencies automatically. This reduces the installation time significantly without having to install modules manually.

4.3 Integrate Perl in Eclipse

EPIC is an Eclipse plugin use for Perl development. A link is provided in the Reference section to the main site which will provide a link to the plugin download page and documentation how-to install and integrate the plugin within Eclipse. On the download page choose the latest EPIC plugin.

4.4. Build UPS Tool in Eclipse using XMLCompile

4.4.1 Create a new Perl project in Eclipse

4.4.2 Import XMLCompileShipConfirmClient.pl to project

4.4.3 Define the values for the following variables accordingly

- access license, userid and password
- endpoint url
- schema file location
  - accessSchemaFile
  - requestSchemaFile
  - responseSchemaFile
  - importedSchemaFile
- outputFileName – response will get save here

4.4.4 Run XMLCompileShipConfirmClient.pl

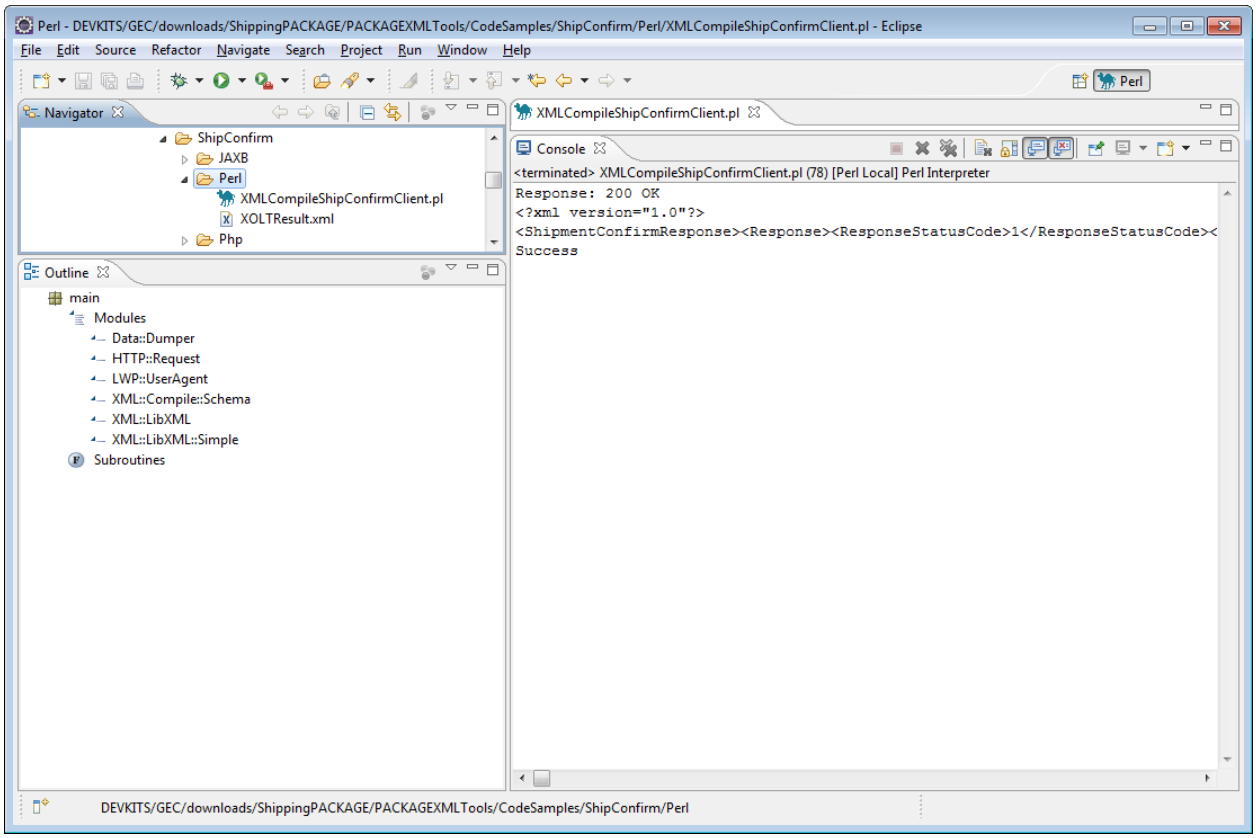


Figure 6: This diagram shows ShipConfirm response message.

UNIX/Linux Development

4.5 Installing Perl

Depending on the setup of your UNIX/Linux operating system, Perl should be already installed on this machine. However, if Perl is not installed on this machine, we strongly recommend consulting with your IT administrator to have this configured.

Note* Must be logged into the system at root level to install Perl.

4.6 Installing Perl modules

There are various tools that can be used like PPM to install Perl modules. In the Reference section we provided a link for information on various tools that can be used for installing Perl modules on UNIX/Linux system.

4.6.1 Firewall/Proxy Configuration

Please consult with your IT administrator to determine if your UNIX/Linux system is behind a firewall that will prevent downloading Perl modules.

4.6.2 Dependencies

To determine what dependencies are needed for UNIX/Linux environment, we strongly recommend consulting with your IT administrator to gather this information before installing Perl and Perl modules.

Note* Must be logged into the system at root level to configure firewall and install Perl module dependencies.

4.7. Build UPS Tool in Eclipse using XMLCompile

4.7.1 Open XMLCompileShipConfirmClient.pl in VI editor
4.7.2 Define the values for the following variables accordingly

- access license, userid and password
- endpoint url
- schema file location
  - accessSchemaFile
  - requestSchemaFile
  - responseSchemaFile
  - importedSchemaFile
- outputFileName – response will get save here

4.7.3 Run XMLCompileShipConfirmClient.pl using the following commands

perl XMLCompileShipConfirmClient.pl

5. Build UPS Tool Sample Code in PHP

- Requirements
- Naming Convention
- UNIX/Linux Development

Requirements

1. PHP 5.3 or above
2. SCA_SDO version 1.2.4 or above

Note* PHP highly recommends building PHP extensions on a UNIX/Linux environments preferably.

Naming Convention

Sample PHP code use following naming convention:
Technology Name + UPSToolName +Client

Example 1: **SDOShipConfirmClient.php**
SDO – Technology name
Ship – UPS Tool Name
Client – Word 'Client' is attached to indicate it is a client program to invoke UPS tool/API.

Example 2: **SDOVoidClient.php**
SDO – Technology name
Void – UPS Tool Name
Client – Word 'Client' is attached to indicate it is a client program to invoke UPS tool/API

- UPS Tool PHP sample code is provided. PHP sample code have the following directory structure:

- Root directory (name of root directory)
  - PHP directory: contains PHP sample client code

UNIX/Linux Development

5.1. Installing PHP

Please consult with your IT administrator about the UNIX/Linux binaries needed for PHP installation.

Note* Must be logged into the system at root level to install PHP.

5.2. Installing SDO Extension

SCA_SDO download link can be found in the Reference section. Please consult with your IT administrator about installing PHP extensions on your UNIX/Linux environment.

5.2.1 Build SDO

A link to php.net site is provided in the Reference section that can help with building and installing PHP extensions on UNIX/Linux system. We recommend using phpize command for simplicity. Please consult with your IT administrator for this process. After the build, the extension will be installed to a specific directory. This directory is mention after the build finishes. This extension contains the SDO program and has to be enabled for PHP to use.

Note* Must be logged into the system at root level to build SDO extension.

5.2.2 Configure PHP.ini

Once the SDO extension has been built successfully, locate and open php.ini in VI editor. Enable the SDO extension by adding the file location where the sdo.so extension is installed.

To verify if the SDO extension was enabled successfully run the following command

php –m

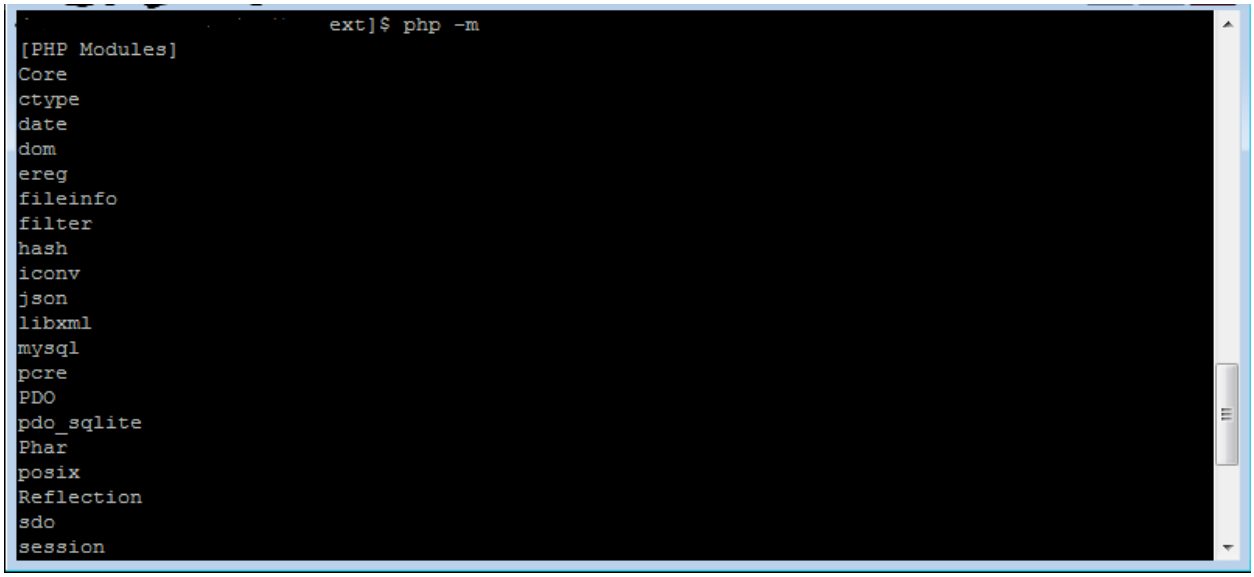


Figure 1: This diagram shows sdo extension enabled

5.2.3 Dependencies

To determine what dependencies are needed for UNIX/Linux environment, we strongly recommend consulting with your IT administrator to gather this information before installing PHP and SDO extension.

5.3 Build UPS Tool in VI using SDO

5.3.1 Open SDOShipConfirmClient.php in VI editor
5.3.2 Define the values for the following variables accordingly

- access license, userid and password
- endpoint url
- schema file location
  - accessSchemaFile
  - requestSchemaFile
  - responseSchemaFile
- outputFileName – response will get save here

5.3.3 Save file
5.3.4 Run SDOShipConfirmClient.php using the following command

php SDOShipConfirmClient.php

6. Perl References

Download site for ActivePerl distribution:
http://www.activestate.com/activeperl/downloads

Tools use for installing Perl modules: http://www.cpan.org/modules/INSTALL.html

XML::Compile module reference site: http://search.cpan.org/~markov/XML-Compile-1.22/lib/XML/Compile.pod

XML::Compile::Schema module reference site: http://search.cpan.org/~markov/XML-Compile-1.22/lib/XML/Compile/Schema.pod

XML::LibXML::Simple module reference site: http://search.cpan.org/~markov/XML-LibXML-Simple-0.91/lib/XML/LibXML/Simple.pod

EPIC plugin tutorial site: http://www.epic-ide.org/

EPIC plugin download site: http://sourceforge.net/projects/e-p-i-c/files/e-p-i-c/

Perl repository site to download and install XML::Pastor:
http://trouchelle.com/perl/ppmrepview.pl

Installing Perl modules (Manual Process) - http://www.thegeekstuff.com/2008/09/how-to-install-perl-modules-manually-and-using-cpan-command/

7. PHP References

Build and compile PHP extensions: http://www.php.net/manual/en/install.pecl.phpize.php

SDO package download site: http://pecl.php.net/get/SCA_SDO

SDO package reference: http://www.php.net/manual/en/sdo-das-xml.examples.php