

Ludovic Rousseau's blog

My activities related to smart card and Free Software (as in free speech).

Sunday, January 11, 2015

PCSC sample in PHP5

In PC/SC sample in different languages I "promised" to give the implementation of the same sample program in many different programming languages.

SCardSCR

In a previous article "[PCSC sample in PHP](#)" I wrote about SCardSCR.

SCardSCR is a project for Windows by Johann Dantant. The [project web site](#) is no more available because the domain name [www.waazaa.org](#) do not exist any more.

A user reported the problem to me and asked where the web site was available. I used a web search service to find where the project SCardSCR had been moved but I found a new PHP PC/SC wrapper instead.

PC/SC for PHP

PC/SC for PHP is the continuation of SCardSCR by Marco Schuster and Johannes Findeisen and is licensed under the PHP license:

This code is licensed under the terms of the PHP License version 3.01. PCSC-Lite is licensed in a way where it is possible to integrate it native in the PHP environment.

The project is hosted at <http://hanez.org/pcsc-for-php.html> and is also available at <http://pecl.php.net/package/pcsc>.

From the project About section:

This is the only extension for using [PC/SC](#) based smart cards with [PHP](#). It is a wrapper to the wonderful and free project [PCSC-Lite](#) which is the middleware to access a smart card using SCard API (PC/SC). Since PCSC-Lite is compatible to the winscard API it should be possible to compile this extension using a Windows(R) operating system. Currently I only take focus on Linux environments.

Thanks are going to Johann Dantant! He provides a PC/SC extension for PHP since 2005 and I reused some of his code. He allowed me to relicense these parts under the terms of the PHP license so I could integrate PCSC-Lite natively into PHP. You find his work [here](#).

Latest version is 0.3 released 2014-04-23

Since the project is hosted inside the PECL (PHP Extension Community Library) I guess/hope the "PC/SC for PHP" project will not disappear soon. From <http://pecl.php.net/>

PECL is a repository for PHP Extensions, providing a directory of all known extensions and hosting facilities for downloading and development of PHP extensions.

API

The API is documented at [PC/SC for PHP - An extension for PHP using the winscard PC/SC API](#).

I note some missing PC/SC features:

- Important missing PC/SC functions: [SCardControl\(\)](#), [SCardGetStatusChange\(\)](#), [SCardBeginTransaction\(\)](#), [SCardEndTransaction\(\)](#)
- Less important missing PC/SC functions: [SCardGetAttrib\(\)](#), [SCardSetAttrib\(\)](#), [SCardListReaderGroups\(\)](#)
- No possibility to indicate the protocol (T=0 or T=1) when connecting to the smart card

Bug

I had to patch the code to add support of T=0 cards.

```
Index: pcsc.c
```

```
=====
```

Google+ Badge

Ludovic Rousseau blog



Follow

Blog Archive

- 2017 (33)
- 2016 (49)
- ▼ 2015 (51)
 - December (8)
 - November (6)
 - October (4)
 - September (3)
 - August (5)
 - July (1)
 - June (4)
 - May (3)
 - April (3)
 - March (2)
 - February (5)
 - ▼ January (7)
 - PCSC sample in Common Lisp
 - PCSC sample in PHP5
 - OS X Yosemite bug: T=0 is used instead of T=1 on d...
 - OS X Yosemite bug: SCARD_E_PROTO_MISMA TCH not retu...
 - OS X Yosemite bug: PC/SC functions crash after a f...
 - OpenSC-devel mailing list statistics for 2014
 - MUSCLE mailing list statistics for 2014
- 2014 (61)
- 2013 (38)
- 2012 (27)
- 2011 (46)
- 2010 (55)

Search This Blog

Subscribe To

Posts

Comments

Google+ Followers

```

--- pcsc.c (révision 335717)
+++ pcsc.c (copie de travail)
@@ -590,7 +590,7 @@ PHP_FUNCTION(scard_list_readers)
    Return a handle to the card */
    PHP_FUNCTION(scard_connect)
    {
-   DWORD dwPreferredProtocol = SCARD_PROTOCOL_T1;
+   DWORD dwPreferredProtocol = SCARD_PROTOCOL_T0 | SCARD_PROTOCOL_T1;
        DWORD dwCurrentProtocol;
        SCARDHANDLE hCard = 0;
        LONG rc = 0;

```

Installation

The installation on Debian is quiet easy.

I had to install the Debian packages:

- [php-pear](#) to get the `pecl(1)` command
- [php5-dev](#) to get the `phpize(1)` command (prepare a PHP extension for compiling)

```

$ sudo pecl install pcsc-alpha
downloading pcsc-0.3.tgz ...
Starting to download pcsc-0.3.tgz (8,939 bytes)
.....done: 8,939 bytes
4 source files, building
running: phpize
Configuring for:
PHP Api Version:      20131106
Zend Module Api No:   20131226
Zend Extension Api No: 220131226
building in /tmp/pear/temp/pear-build-rootMiAGhV/pcsc-0.3
running: /tmp/pear/temp/pcsc/configure
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for a sed that does not truncate output... /bin/sed

[...]

libtool: finish: PATH="/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/sbin" ldconfig -n /tmp/pear/temp/pear-build-rootMiAGhV/pcsc-0.3/modules
-----
Libraries have been installed in:
  /tmp/pear/temp/pear-build-rootMiAGhV/pcsc-0.3/modules

If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the '-LLIBDIR'
flag during linking and do at least one of the following:
- add LIBDIR to the 'LD_LIBRARY_PATH' environment variable
  during execution
- add LIBDIR to the 'LD_RUN_PATH' environment variable
  during linking
- use the '-Wl,-rpath -Wl,LIBDIR' linker flag
- have your system administrator add LIBDIR to '/etc/ld.so.conf'

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
-----

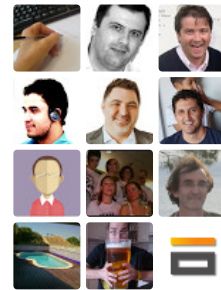
Build complete.
Don't forget to run 'make test'.

running: make INSTALL_ROOT="/tmp/pear/temp/pear-build-rootMiAGhV/install-pcsc-0.3" install
Installing shared extensions:      /tmp/pear/temp/pear-build-rootMiAGhV/install-pcsc-0.3/usr/lib/php5/20131226/
Installing header files:           /tmp/pear/temp/pear-build-rootMiAGhV/install-pcsc-0.3/usr/include/php5/
running: find "/tmp/pear/temp/pear-build-rootMiAGhV/install-pcsc-0.3" | xargs ls -dils

```

Ludovic Rousseau b...

Follow



336 have us in circles

[View all](#)

```

305074 4 drwxr-xr-x 3 root root 4096 janv. 11 14:21 /tmp/pear/temp/pear-build-rootMi
AGhV/install-pcsc-0.3
305078 4 drwxr-xr-x 4 root root 4096 janv. 11 14:21 /tmp/pear/temp/pear-build-rootMi
AGhV/install-pcsc-0.3/usr
305082 4 drwxr-xr-x 3 root root 4096 janv. 11 14:21 /tmp/pear/temp/pear-build-rootMi
AGhV/install-pcsc-0.3/usr/include
305083 4 drwxr-xr-x 2 root root 4096 janv. 11 14:21 /tmp/pear/temp/pear-build-rootMi
AGhV/install-pcsc-0.3/usr/include/php5
276538 4 -rw-r--r-- 1 root root 1238 janv. 11 14:21 /tmp/pear/temp/pear-build-rootMi
AGhV/install-pcsc-0.3/usr/include/php5/php_pcsc.h
305079 4 drwxr-xr-x 3 root root 4096 janv. 11 14:21 /tmp/pear/temp/pear-build-rootMi
AGhV/install-pcsc-0.3/usr/lib
305080 4 drwxr-xr-x 3 root root 4096 janv. 11 14:21 /tmp/pear/temp/pear-build-rootMi
AGhV/install-pcsc-0.3/usr/lib/php5
305081 4 drwxr-xr-x 2 root root 4096 janv. 11 14:21 /tmp/pear/temp/pear-build-rootMi
AGhV/install-pcsc-0.3/usr/lib/php5/20131226
276304 60 -rwxr-xr-x 1 root root 60896 janv. 11 14:21 /tmp/pear/temp/pear-build-rootMi
AGhV/install-pcsc-0.3/usr/lib/php5/20131226/pcsc.so

```

```

Build process completed successfully
Installing '/usr/lib/php5/20131226/pcsc.so'
Installing '/usr/include/php5/php_pcsc.h'
install ok: channel://pecl.php.net/pcsc-0.3
configuration option "php_ini" is not set to php.ini location
You should add "extension=pcsc.so" to php.ini

```

I had to edit the file `/etc/php5/cli/php.ini` to change the `enable_dl` definition:

```

; Whether or not to enable the dl() function. The dl() function
; does NOT work properly in multithreaded servers, such as IIS or Zeus,
; and is automatically disabled on them.
; http://php.net/enable-dl
;enable_dl = Off
enable_dl = On

```

Source code

```

<?php

if (!extension_loaded('pcsc')) {
    dl('pcsc.so');
}

# Get a PC/SC context
$context = scard_establish_context();
//var_dump($context);

# Get the reader list
$readers = scard_list_readers($context);
//var_dump($readers);

# Use the first reader
$reader = $readers[0];
echo "Using reader: ", $reader, "\n";

# Connect to the card
$connection = scard_connect($context, $reader);
//var_dump($connection);

# Select Applet APDU
$CMD = "00A40400AA0000006203010C0601";
$res = scard_transmit($connection, $CMD);
var_dump($res);

# test APDU
$CMD = "00000000";
$res = scard_transmit($connection, $CMD);
var_dump($res);
echo pack("H*", $res), "\n";

```

```
# Release the PC/SC context
scard_release_context($context);

?>
```

Output

```
$ php sample.php
Using reader: Gemalto PC Twin Reader 00 00
string(4) "9000"
string(28) "48656C6C6F20776F726C64219000"
Hello world!💎
```

Conclusion

I have only tested the wrapper using the command line php5 program. I guess the wrapper should also be available from a PHP script in a HTML page hosted by a web server. In that case the PC/SC commands are executed on the server. I don't know what it can be used for. In general you want to use the smart card on the client side.

Happy hacking with this PHP PC/SC wrapper.



Labels: [code](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

Bitcoin



License: [by-nc-sa](#)



This blog by [Ludovic Rousseau](#) is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](#).

Simple theme. Powered by [Blogger](#).