

# PKCS#11 Wrapper for Java

from IAIK

<http://jce.iaik.tugraz.at>

Version 1.4

6 March 2015

## **Introduction**

## Introduction

This is a library to access PKCS#11 modules from the Java programming language [3]. It uses the Java Native Interface [4] to access the PKCS#11 modules of smart cards or other hardware security modules (HSM). People at IBM had the idea of implementing such a wrapper much earlier. Their wrapper [5] also works very well, but one cannot use their implementation for commercial purposes or redistribute it for any other purpose.

Please notice that this library does not come with a JCA or JCE provider implementation. For this purpose there is a different product – the IAIK PKCS#11 Provider [1].

The documentation of this library assumes that the reader is familiar with the basic principles of PKCS#11. There is a general overview chapter in the PKCS#11 specification from RSA Laboratories [2]. It gives a brief introduction into the basics of PKCS#11.

## The Layer Model of the System

Figure 1 shows the layer model of this library. This library consists of the Object Oriented (OO) Java Wrapper API for PKCS#11, the (non-Object Oriented) Java Wrapper API for PKCS#11 and the Native Module of the Wrapper, the green layers in the figure. The following paragraphs describe these parts. The lowest layer, the PKCS#11 Module of the Smart Card, is the PKCS#11 module that the smart card manufacturer supplies. This is normally a DLL or shared library. As the arrows show, the uppermost layer depends on the Java Wrapper for PKCS#11, but not vice versa. This means you can use the Java Wrapper for PKCS#11 directly and build your application upon it without using the OO layer. This can be useful to create smaller applications, because you do not need most of the classes of the package `iaik.pkcs.pkcs11` and no class from `iaik.pkcs.pkcs11.objects` and `iaik.pkcs.pkcs11.parameters`. The only classes from `iaik.pkcs.pkcs11` you need are the exception classes.

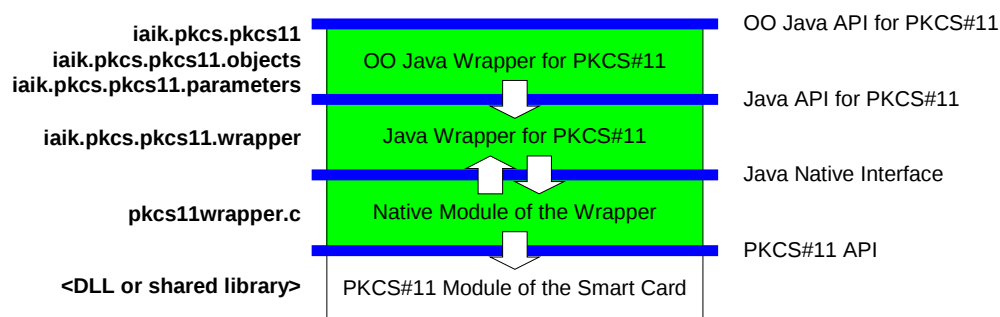


Figure 1

## ***The Object-Oriented Java API for PKCS#11***

This object-oriented Java API resides in the packages `iaik.pkcs.pkcs11`, `iaik.pkcs.pkcs11.objects` and `iaik.pkcs.pkcs11.parameters`. It provides a straight forward mapping of the PKCS#11 v2.20 standard to a set of classes and interfaces. The package `iaik.pkcs.pkcs11.objects` is a model of the object hierarchy presented in this PKCS#11 standard. The package `iaik.pkcs.pkcs11.parameters` provides classes for objects that act as parameters for mechanisms which require specific arguments. This layer solely builds upon the Java API for PKCS#11 as implemented by the Java Wrapper for PKCS#11.

## ***The Java API for PKCS#11***

The non-Object Oriented Java Wrapper API for PKCS#11 is a set of Java classes and interfaces that reflects the PKCS#11 API. It is a straightforward realization of the data structures as defined in PKCS#11. For each structure in the `pkcs11t.h` header file of PKCS#11, there is a corresponding class in the package `iaik.pkcs.pkcs11.wrapper`. Notice, that this is not an object oriented approach at this level; it is just a straightforward mapping of the data structures to Java. All adoptions to the PKCS#11 API, including wrapping into an object oriented approach, appear in the Object Oriented Java Wrapper API for PKCS#11. The interface `PKCS11` in the `iaik.pkcs.pkcs11.wrapper` package is the interface to a PKCS#11 module and provides access to the functions defined by PKCS#11. All names of classes, data structures and methods are the same as the corresponding PKCS#11 counterpart. The `PKCS11Connector` instantiates an object that implements this `PKCS11` interface. The returned object gives access to the PKCS#11 module of the smart card; it is the Java-Counterpart to the `CK_C_GetFunctionList` returned by the `C_GetFunctionList` function in PKCS#11. The `Module` class in the object-oriented layer provides the respective functionality. Have a look at the `demo.pkcs.pkcs11` package in the `demo` directory for sample programs.

## ***The Native Module of the Wrapper***

This native module of the wrapper is responsible for translation of the Java data structures, which the Java API for PKCS#11 part defines, to native PKCS#11 data structures and vice versa. This module of the system does not include any additional logic, it only provides a straightforward mapping from the Java API for PKCS#11 to the PKCS#11 Module of the Smart Card. This layer is necessary, because the JNI requires the native functions to have a special signature that is defined by JNI itself. PKCS#11 and JNI are not compatible as they are, and this is the reason why this layer is necessary at all. In compiled form, this module is a native DLL or shared library.

## ***Includes:***

Full source of all components under an Apache-style license as well as precompiled binaries for Windows, Linux, Solaris 8/9/10 and Mac OS X.

## Requirements:

JDK 1.3 or higher, GNU make for building the project using the included make files, MS VC++ 6.0 for building the DLLs for Windows, GCC for building the shared libraries for Linux and Solaris (or SUN C compiler). Some of the included demo programs require the IAIK-JCE to compile and run. Simply place the iaik\_jce\_full.jar in the examples\lib directory.

## License

We provide this software under an Apache style license. The complete license text is as follows.

Copyright (c) 2002 Graz University of Technology. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by IAIK of Graz University of Technology."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Graz University of Technology" and "IAIK of Graz University of Technology" must not be used to endorse or promote products derived from this software without prior written permission.
5. Products derived from this software may not be called "IAIK PKCS Wrapper", nor may "IAIK" appear in their name, without prior written permission of Graz University of Technology.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE LICENSOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY

WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## References

- [1] IAIK Java Cryptography Toolkits,  
<http://jce.iaik.tugraz.at/>
- [2] PKCS#11, Version 2.11 and Version 2.20,  
<http://www.cryptsoft.com/pkcs11doc/>
- [3] Java 3 Platform, by Oracle,  
<http://www.oracle.com/technetwork/java/javase>
- [4] Java Native Interface 1.3, by Oracle,  
<http://docs.oracle.com/javase/7/docs/technotes/guides/jni/>
- [5] PKCS#11 API for Java, by IBM Alphaworks,  
<http://alphaworks.ibm.com/>