SECURITY

Overview

SECURITY - java.security (some classes in KERNEL), java.security.acl, java.security.cert, java.security.interfaces, java.security.spec, javax.security.cert.

Status

Check the component_development_status page for the status of other ClassLibrary modules.

J2SE 1.4.2

The current codes cover J2SE 1.4.2 API

Java 5.0

Current code is mostly compliant with J2SE 1.5 specification. It should be updated to reflect new Java 5.0 enhancements (e.g. generics, enums) when harmony has a 1.5 compatible VM

Below are some details about what's available, missing and in progress

Public API

This component contains implementation of public API, internal code and unit tests. All public 1.5 classes are implemented and tested..

The code contains a number of TODO's such as:

- switch to generics and/or enums
- use more efficient API introduced in 1.5
- implement optional features

Security Providers

Implementation of Certificate Factory based on internal ASN.1 functionality is included.

Provider name is "DRLCertFactory" (to be renamed)

Certificate Factory implementation is able to generate X.509 Certificates, CRLs, and CertPath objects on the base of their encoded forms. Supported certificate type is "X.509". This type should be used when generating an instance of CertificateFactory.

The supported encoded forms of certificates are

- ASN.1 DER encoded form (as specified by RFC 3280)
- PEM Encoded form, i.e. Base64 encoded form of ASN.1 encoding

The supported encoded forms of CRLs are

- ASN.1 DER encoded form (as specified by RFC 3280)
- PEM Encoded form, i.e. Base64 encoded form of ASN.1 encoding

The supported encoded forms of CertPath objects are:

PkiPath encoded form, i.e. sequence of ASN.1 DER encoded certificates (ASN.1 definition is PkiPath ::= SEQUENCE OF Certificate) CertPath can be generated from PKCS7 SignedData object provided in the form of ASN.1 DER encoded ContentInfo structure. Factory retrieves SignedData structure from ContentInfo structure and generated CertPath object represents the information presented in ‘certificates’ field of the SignedData object.

No other providers are currently implemented.

When crypto functionality is necessary one may use open source BouncyCastle provider

Known issues:

Signed jar file verification requires some crypto algorithms. But BouncyCastle provider is located in a signed jar file. So, at least Message Digest SHA-1 and Signature SHA1withDSA should be implemented and placed into unsigned jar to verify signature in BouncyCastle jar. The code contains simplified implementation of secure random algorithm. SHA1PRNG algorithm should be implemented

Tools

No security related tools (e.g. keytool, jarsigner) are currently implemented.