Contents

1. Client-Side API for Java Card RMI  1
   1.1 Basic Features  1
   1.2 Client Application Interfaces  2
       1.2.1 CardAccessor Interface  2
       1.2.2 JavaCardRMICConnect Class  2
       1.2.3 Securing the Transport Layer  3
   1.3 Stub/Proxy Generation  4
       1.3.1 CardObjectFactory Class  4
   1.4 Exceptions  5

2. Client-Side Java Card RMI Framework API  7
   2.1 Package rmiclient  7
       2.1.1 Exception Classes  8
   2.2 API Documentation  8
       Overview 11
       com.sun.javacard.javax.smartcard.rmiclient 13
       javacard.framework 43
       javacard.framework.service 63
       javacard.security 67

Index  71
Client-Side API for Java Card RMI

A Java Card RMI client application runs on a Card Acceptance Device (CAD) terminal that supports a J2SE or J2ME platform. The client application requires a portable and platform independent mechanism to access the Java Card RMI (JCRMI) server applet executing on the smart card.

This client side architecture currently focuses on a J2SE client running on a JDK1.2 or higher environment. The client interfaces are designed to be independent of the card access framework. The interfaces therefore only provide JCRMI specific access mechanisms and allow the client application to use its preferred card access mechanisms for its basic card interaction needs.

The example in Sun Microsystem’s reference implementation uses the Open Card Framework for its card access mechanisms. See the Java Card™ 2.2 Development Kit User’s Guide provided with this release.

The Open Card Framework classes provide a Java application platform independent access to a connected smart card.

The J2ME environment is more limited, and since most J2ME profiles do not support the JCRMI machinery at this time, it is not discussed in detail. The design of the JCRMI client-side architecture ensures that it does not preclude the J2ME client.

1.1 Basic Features

The Java Card RMI client side architecture caters to the following:

- Provides the client application mechanisms to initialize and initiate an RMI session with a Java Card applet.
- Provides the client application access to the initial remote reference.
- Allows the client application to customize the message packet during Java Card communication to introduce transport level security.
Allows the CAD terminal developer to customize the card access mechanisms for the specifics of the communications hardware.

Allows the CAD terminal developer to customize the stub generation mechanism for one that is best suited to the platform capabilities.

1.2 Client Application Interfaces

These classes provide the basic Java Card RMI client application with the front-end mechanisms to connect to the Java Card applet and obtain an initial reference to invoke remote methods.

As mentioned earlier these client interfaces are designed to be independent of the card access framework. The examples referenced use Open Card Framework to build a fully portable client application.

1.2.1 CardAccessor Interface

The CardAccessor interface is used by the JCRMI client framework to communicate with the card. It defines the following methods:

- public byte[] exchangeAPDU( byte[] command )

  The JCRMI client framework uses this method to send a command APDU to the card and receive the response APDU.

- public short getSessionIdentifier()

  This method returns an identifier associated with the current card session.

The client platform must provide the client application with a class that implements the CardAccessor interface and the means to obtain an instance.

In the JCRMI example provided in this release (see Chapter 3 of the Java Card™ 2.2 Development Kit User’s Guide), the client application interacts with the OCF classes to obtain an instance of the OCFCardAccessor class that implements the CardAccessor interface.

1.2.2 JavaCardRMICConnect Class

The client application uses the JavaCardRMICConnect class to initiate the JCRMI session and obtain the initial remote reference from the card.
Constructor:

public JavaCardRMIConection(CardAccessor cA)

The client constructs an instance of the JavaCardRMIConection class, initializing it with a CardAccessor object.

The methods of JavaCardRMIConection are:

- public byte[] selectApplet(byte[] appletAID)
- public byte[] selectApplet(byte[] appletAID, CardObjectFactory cOF)

The client starts a JCRMI session with the Java Card applet on the card using this method. The second parameter is optional and specifies a custom CardObjectFactory.

- public Remote getInitialReference()

To begin a JCRMI based dialog, the client obtains the first remote object reference via this method and then casts it to the appropriate type.

1.2.3 Securing the Transport Layer

A simple Java Card RMI client application would only depend on the above methods for a remote method based card session with a Java Card applet. The Java Card applet might need to layer additional security services to protect the message transport data for integrity and privacy. In order to do this, the client application (or security service provider) could subclass the implementation of the CardAccessor class with its own version and re-define the exchangeAPDU method. The JCRMI example provided in this release shows the SecureOCFCardAccessor class, which subclasses the OCFCardAccessor class for this purpose.

To superimpose security mechanisms on the transport message packet, a client application’s re-implementation of the exchangeAPDU method would access the message data inside the byte array, and filter the data accordingly before calling the super.exchangeAPDU method to continue with the exchange. Similarly, on return from the super method, it would filter the response data, if required, before returning.
1.3 Stub/Proxy Generation

The client does not usually interact with the client side stub generation classes. It is used by the JavaCardRMIClass to instantiate the initial remote stub object and subsequently the stub objects themselves to instantiate other remote stub objects.

The client library classes include a class called CardObjectFactory which encapsulates the mechanism used to instantiate stubs on the client.

The simplest implementation of CardObjectFactory uses the SELECT REF_FORMAT_CLASS and simply locates the pre-compiled stub class corresponding to the implementation class name returned in the remote reference descriptor from the card and instantiates it.

An alternate implementation uses SELECT REF_FORMAT_INTERFACES and the list of remote interfaces implemented by the remote class. The dynamic proxy generation mechanism, defined by JDK1.3, generates the proxy instances.

1.3.1 CardObjectFactory Class

This abstract class allows the encapsulation of the response format of the Java Card RMI protocol. The class also customizes any specific mechanisms used to instantiate remote stub objects on the client.

The key methods of CardObjectFactory are:

- CardObjectFactory ( cu CardAccessor )
  This is the constructor that is initialized with the CardAccessor instance.
- Remote getObject ( byte[] buffer, int tagOffset, Class type )
  This method is called by both the JavaCardRMIClass as well as the remote stub classes themselves to obtain an instance of the card object encoded in the Java Card RMI response format. The object returned may be an instance of a remote stub class associated with the specified remote reference descriptor received from the card, an array of primitives, or a primitive wrapped instance of a Java wrapper class.
- byte setINSByte ( byte ins )
  This method sets the INVOKE command byte to be used during this selection session. This method is called by JavaCardRMIClass after a applet has been selected for JCRMI dialogue.
- byte getINSByte ()

4 RMI Client Application Programming Interface • June, 2002
This method returns the INVOKE command byte to be used during this selection session.

\[
\text{byte getRemoteRefFormat ()}
\]

This abstract method returns the format of the remote object reference descriptor supported by the stub factory implementation. The method returns one of:

\[
\text{REF_FORMAT_NONE, REF_FORMAT_CLASS, or REF_FORMAT_INTERFACES}
\]

\[
\text{protected abstract getRemoteObject ( byte[] buffer, int tagOffset )}
\]

This abstract method instantiates a stub or proxy object corresponding to the remote reference returned from the card.

## 1.4 Exceptions

The remote method invoked on the card may throw an exception to signal that an unexpected condition has been detected. The RMIService class on the card catches the exception and returns this information to the client.

If the exception thrown on the card is an exception defined in the Java Card 2.2 API, the same exception is re-thrown by the stub object back to the client application. The client can access the reason code associated with Java Card-specific exceptions using the standard `getReason()` method.

If the exception thrown on the card is a subclass of an exception defined in the Java Card 2.2 API, a client subclass of the closest exception defined in the API (along with the reason code, if applicable) is re-thrown to the client by the stub object. The exception object will contain the following error message string: “A subclass was thrown on the card.”

Apart from the exceptions thrown by the remote method itself, errors during communication, marshalling, protocol, unmarshalling, stub generation and so forth related to the JCRMI method invocation results in a `RemoteException` exception being thrown to the client application with appropriate error message strings which include the following:

- Applet selection failed, \( SW = \ldots \)
- Cannot create default CardObjectFactory
- Incorrect (too short) response received from the card
- Invalid format requested by the factory
- Invoke operation failed, \( SW = \ldots \)
- Method not found
- Object not exported
- Out of param resources
- Out of response resources
- Protocol error reported by the card
- Signature mismatch
- Thrown on the card
- Unexpected exception received while instantiating the stub
- Unsupported APPLICATION tag
- Unsupported error type received from the card
- Unsupported exception type received from the card
- Unsupported FCI tag
- Unsupported JCRMI tag
- Unsupported JCRMI version
- Unsupported subclass exception type received from the card
- Unsupported tag
- Wrong parameter to getNonRemoteObject()
CHAPTER 2

Client-Side Java Card RMI Framework API

This chapter begins with an overview of the classes in the Java Card RMI Client package, com.sun.javacard.jsa.nax.javascriptcard.rmiclient. Then follows the API documentation generated by the Javadoc tool for this package.

2.1 Package rmiclient

In the package com.sun.javacard.jsa.nax.javascriptcard.rmiclient:

- abstract class CardObjectFactory

  Key methods in this class are getObject and getRemoteObject:

  public Object getObject( byte[] buffer, int tagOffset, Class type ) throws RemoteException, StubNotFoundException, Exception

  This is a public method which instantiates an object or throws an exception corresponding to the card response. This method is used to parse card responses to select and during method invocations to parse card responses. If the response is a remote reference, the method getRemoteObject is called. Generally, an implementor do not need to override the getObject method.

  protected abstract Remote getRemoteObject(byte[] buffer, int tagOffset) throws Exception;

  This is an abstract method intended to parse card responses containing remote references (note: this method does not have a type parameter). Because there are two different reference formats (with class name and with interface names), an implementor must override this abstract method with an implementation that parses the format used by the application.
interface CardAccessor

Declares method exchangeAPDU(), exchanging data with the card. 
Implementations of this interface usually perform additional transformations of 
the data, such as encryption or integrity checks.

Declares methodgetSessionIdentifier().

class JavaCardRMICConnect

Provides methods selectApplet() and getInitialReference() for the 
client program.

The CardAccessor instance must be provided to the constructor of this class.

A custom CardObjectFactory can optionally be specified for the selectApplet 
method. If the ObjectFactory is not specified, the current implementation 
JavaCardRMICConnect uses 
com.sun.javacard.ocfrmiclientimpl.JCCardObjectFactory as default.

2.1.1 Exception Classes

Exceptions contained in this package:
APDUExceptionSubclass.java
CardExceptionSubclass.java
CardRuntimeExceptionSubclass.java
CryptoExceptionSubclass.java
PINExceptionSubclass.java
ISOExceptionSubclass.java
ServiceExceptionSubclass.java
UserExceptionSubclass.java
TransactionExceptionSubclass.java

These exception classes are subclasses of the corresponding Java Card exceptions 
and are designed to be re-thrown on the client when a subclass of the corresponding 
exception is thrown on the card by a native method.

2.2 API Documentation

The remainder of this chapter contains the API documentation generated by the 
Javadoc tool for of the client-side JCRMI application programming interface 
package, com.sun.javacard.javax.smartcard.rmiclient, including 
subclasses of Java Card exceptions.
This chapter also includes Java Card exceptions that can be re-thrown from the client. These exceptions reside in the Java Card packages `javacard.framework`, `javacard.framework.service`, and `javacard.security`. 
Overview

Package Summary

<table>
<thead>
<tr>
<th>Packages</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.sun.javacard.javasmartcard.rmiclient</td>
<td>Provides framework of classes and interfaces for a Java Card client.</td>
</tr>
<tr>
<td>javacard.framework</td>
<td>Provides Java Card exceptions that may be re-thrown on the client.</td>
</tr>
<tr>
<td>javacard.framework.service</td>
<td>Provides Java Card exceptions that may be re-thrown on the client.</td>
</tr>
<tr>
<td>javacard.security</td>
<td>Provides Java Card exceptions that may be re-thrown on the client.</td>
</tr>
</tbody>
</table>

Class Hierarchy

java.lang.Object
  com.sun.javacard.javasmartcard.rmiclient.CardObjectFactory
  com.sun.javacard.javasmartcard.rmiclient.JavaCardRMIConnect
  java.lang.Throwable (implements java.io.Serializable)
    javacard.framework.CardException
      com.sun.javacard.javasmartcard.rmiclient.CardExceptionSubclass
    javacard.framework.UserException
      com.sun.javacard.javasmartcard.rmiclient.UserExceptionSubclass
    javacard.security
      java.lang.RuntimeException
        javacard.framework.CardRuntimeException
          javacard.framework.APDUException
            com.sun.javacard.javasmartcard.rmiclient.APDUExceptionSubclass
          javacard.security.CryptoException
            com.sun.javacard.javasmartcard.rmiclient.CryptoExceptionSubclass
          javacard.framework.ISOException
            com.sun.javacard.javasmartcard.rmiclient.ISOExceptionSubclass
          javacard.framework.PINException
            com.sun.javacard.javasmartcard.rmiclient.PINExceptionSubclass
          javacard.framework.service.ServiceException
            com.sun.javacard.javasmartcard.rmiclient.ServiceExceptionSubclass
          javacard.framework.SystemException
            com.sun.javacard.javasmartcard.rmiclient.SystemExceptionSubclass
          javacard.framework.TransactionException
            com.sun.javacard.javasmartcard.rmiclient.TransactionExceptionSubclass

TransactionExceptionSubclass

Interface Hierarchy

com.sun.javacard.javasmartcard.rmiclient.CardAccessor
Package

com.sun.javacard.javax.smartcard.rmiclient

Description
Provides framework of classes and interfaces for a Java Card client.

Class Summary

<table>
<thead>
<tr>
<th>Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CardAccessor</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CardObjectFactory</strong></td>
</tr>
<tr>
<td><strong>JavaCardRMIConnect</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exceptions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>APDUExceptionSubclass</strong></td>
</tr>
<tr>
<td><strong>CardExceptionSubclass</strong></td>
</tr>
<tr>
<td><strong>CardRuntimeExceptionSubclass</strong></td>
</tr>
<tr>
<td><strong>CryptoExceptionSubclass</strong></td>
</tr>
<tr>
<td><strong>ISOExceptionSubclass</strong></td>
</tr>
<tr>
<td><strong>PINExceptionSubclass</strong></td>
</tr>
<tr>
<td><strong>ServiceExceptionSubclass</strong></td>
</tr>
<tr>
<td><strong>SystemExceptionSubclass</strong></td>
</tr>
<tr>
<td><strong>TransactionExceptionSubclass</strong></td>
</tr>
<tr>
<td><strong>UserExceptionSubclass</strong></td>
</tr>
</tbody>
</table>
Declaration

public class APDUExceptionSubclass extends APDUException

java.lang.Object
   |--java.lang.Throwable
      |--java.lang.Exception
         |--java.lang.RuntimeException
            |--javacard.framework.CardRuntimeException
               |--javacard.framework.APDUException
                  |--com.sun.javacard.javax.smartcard.rmi.client.APDUExceptionSubclass

All Implemented Interfaces: java.io.Serializable

Description

This exception class represents a card subclass of APDUException. APDUException represents an APDU related exception on the card.

Member Summary

Constructors

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APDUExceptionSubclass(java.lang.String msg, short reason)</td>
<td>Constructs an APDUExceptionSubclass with the specified reason and specified error message.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.lang.String getMessage()</td>
<td>Returns the error message string of this throwable object.</td>
</tr>
</tbody>
</table>

Inherited Member Summary

Fields inherited from interface APDUException

BAD_LENGTH, BUFFER_BOUNDS, ILLEGAL_USE, IO_ERROR, NO_T0_GETRESPONSE, NO_T0_REISSUE, T1_IFD_ABORT

Methods inherited from interface APDUException

throwIt(short)
Inherited Member Summary

Methods inherited from interface CardRuntimeException
getReason(), setReason(short)

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

Methods inherited from class java.lang.Throwable
fillInStackTrace, getLocalizedMessage, printStackTrace, printStackTrace, toString

Constructors

APDUExceptionSubclass(String, short)

public APDUExceptionSubclass(java.lang.String msg, short reason)

Constructs an APDUExceptionSubclass with the specified reason and specified error message.

Parameters:

msg - the associated message string
reason - the reason for the exception.

Methods

getMessage()

public java.lang.String getMessage()

Returns the error message string of this throwable object.

Overrides: getMessage in class Throwable

Returns: the error message string of this Throwable object if it was created with an error message string; or null if it was created with no error message.
CardAccessor

Description
The CardAccessor interface represents a generic smartcard communication API. The interface based definition makes it platform and framework independent. This interface is used by Java Card RMI stubs to access the smart card.

Member Summary

<table>
<thead>
<tr>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte[] exchangeAPDU(byte[] sendData)</td>
</tr>
<tr>
<td>short getSessionIdentifier()</td>
</tr>
</tbody>
</table>

Methods

exchangeAPDU(byte[])

```java
public byte[] exchangeAPDU(byte[] sendData)
    throws IOException
```

This method sends the specified data to the smartcard, waits for the response and returns the response in the return data. The input data is assumed to be formatted for ISO 7816-4 APDU communication as follows: [0] = CLA, [1] = INS, [2] = P1, [3] = P2, [4] = Lc, [4..] = command data. The response data is formatted for ISO 7816-4 APDU communication as follows: [0] = SW1, [1] = SW2, [2..] = response data.

Parameters:
sendData - the ISO 7816-4 formatted command APDU data with 5 bytes of header followed by the command data.

Returns: responseData contains the response received from card with the 2 status bytes followed by the response data.

Throws:
    IOException - if communication error occurs

gSessionIdentifier()

```java
public short getSessionIdentifier()
```

This method returns an identifier associated with the card session. A new session identifier is assigned on every card reset.
**Returns:** the card session identifier
Declaration
public class CardExceptionSubclass extends CardException

java.lang.Object
   |---java.lang.Throwable
      |---java.lang.Exception
         |---javacard.framework.CardException
            |---com.sun.javacard.javax.smartcard.rmiclient.CardExceptionSubclass

All Implemented Interfaces: java.io.Serializable

Description
This exception class represents a subclass of CardException on the card. The CardException class on the card defines a field reason and two accessor methods getReason() and setReason(). The reason field encapsulates exception cause identifier in Java Card. All Java Card checked Exception classes on the card should extend CardException.

<table>
<thead>
<tr>
<th>Member Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructors</td>
</tr>
<tr>
<td>CardExceptionSubclass(java.lang.String msg, short reason)</td>
</tr>
</tbody>
</table>
Construct a CardExceptionSubclass instance with the specified reason and the specified error message.

<table>
<thead>
<tr>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.lang.String getMessage()</td>
</tr>
</tbody>
</table>
Returns the error message string of this throwable object.

Inherited Member Summary
Methods inherited from interface CardException
getReason(), setReason(short), throwIt(short)

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Methods inherited from class java.lang.Throwable
### CardExceptionSubclass

**CardExceptionSubclass(String, short)**

```java
public CardExceptionSubclass(java.lang.String msg, short reason)
```

Construct a CardExceptionSubclass instance with the specified reason and the specified error message.

**Parameters:**
- `msg` - the associated message string
- `reason` - the reason for the exception

### Methods

**getMessage()**

```java
public java.lang.String getMessage()
```

Returns the error message string of this throwable object.

**Overrides:** `getMessage` in class Throwable

**Returns:** the error message string of this Throwable object if it was created with an error message string; or null if it was created with no error message.
Declaration

public abstract class CardObjectFactory

java.lang.Object


Description

This CardObjectFactory abstract class represents the base class for Java Card 2.2 RMI stub generation implementations. An instance of this class is associated with one Java Card applet selection session.

Member Summary

<table>
<thead>
<tr>
<th>Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected CardAccessor cardAccessor</td>
</tr>
<tr>
<td>the CardAccessor to be used for the current RMI session.</td>
</tr>
</tbody>
</table>

| static byte REF_FORMAT_CLASS |
| This value (=1) is used to signify that the CardObjectFactory implementation supports the JCRMI remote reference format using the name of the card implementation remote class. |

| static byte REF_FORMAT_INTERFACES |
| This value (=2) is used to signify that the CardObjectFactory implementation supports the JCRMI remote reference format using the names of the remote interfaces implemented by the card implementation remote class. |

| static byte REF_FORMAT_NONE |
| This value (=0) is used to signify that the CardObjectFactory implementation does not support any JCRMI remote reference descriptor formats. |

| Constructors |
| CardObjectFactory(CardAccessor cA) |
| Creates a new CardObjectFactory for this RMI session |

| Methods |
| byte getINSByte() |
| returns the configured ISO 7816-4 command INS byte to be used in the Java Card remote method invocation command |

| java.lang.Object getObject(byte[] buffer, int tagOffset, java.lang.Class type) |
| This abstract method returns the instance of the card object corresponding to the value returned from the card. |

| protected abstract java.rmi.Remote getRemoteObject(byte[] buffer, int tagOffset) |
| This abstract method instantiates a stub or proxy object corresponding to the remote reference returned from the card |

| abstract byte getRemoteRefFormat() |
| returns the format of JCRMI remote object reference descriptor supported. |

| void setINSByte(byte ins) |
| Sets the ISO 7816-4 header INS byte to use for Java Card RMI method invocation commands. |
Inherited Member Summary

Methods inherited from class java.lang.Object
class clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Fields

REF_FORMAT_NONE

public static final byte REF_FORMAT_NONE

This value (=0) is used to signify that the CardObjectFactory implementation does not support any JCRMI remote reference descriptor formats.

REF_FORMAT_CLASS

public static final byte REF_FORMAT_CLASS

This value (=1) is used to signify that the CardObjectFactory implementation supports the JCRMI remote reference format using the name of the card implementation remote class.

REF_FORMAT_INTERFACES

public static final byte REF_FORMAT_INTERFACES

This value (=2) is used to signify that the CardObjectFactory implementation supports the JCRMI remote reference format using the names of the remote interfaces implemented by the card implementation remote class.

cardAccessor

protected com.sun.javacard.java.smartcard.rmiClient.CardAccessor cardAccessor

the CardAccessor to be used for the current RMI session.

Constructors

CardObjectFactory(CardAccessor)

public CardObjectFactory(com.sun.javacard.java.smartcard.rmiClient.CardAccessor cA)

Creates a new CardObjectFactory for this RMI session

Parameters:
cA - The CardAccessor for the current session.
Methods

**getObject(byte[], int, Class)**

```java
public java.lang.Object getObject(byte[] buffer, int tagOffset, java.lang.Class type)
    throws RemoteException, StubNotFoundException, Exception
```

This abstract method returns the instance of the card object corresponding to the value returned from the card. The value returned by the card may be any of the valid return types in Java Card 2.2 RMI protocol format. In particular the return type may be a primitive, an array type or a remote reference descriptor or null. The method delegates processing to getRemoteObject(...) when the response tag is normal and expected type is a remote reference.

**Parameters:**
- `buffer` - the byte array containing the JC 2.2 RMI response data
- `tagOffset` - the offset within the array where the response tag is located
- `type` - the expected return type in the response data

**Returns:** the object associated with the response data. Primitive return value is encapsulated in a wrapper object.

**Throws:**
- `StubNotFoundException` - if an appropriate object could not be instantiated
- `RemoteException` - for any RMI-specific exceptions
- `Exception` - for exceptions thrown on the card

**getRemoteObject(byte[], int)**

```java
protected abstract java.rmi.Remote getRemoteObject(byte[] buffer, int tagOffset)
    throws Exception
```

This abstract method instantiates a stub or proxy object corresponding to the remote reference returned from the card.

**Parameters:**
- `buffer` - the byte array containing the JC 2.2 RMI response data
- `tagOffset` - the offset within the array where the response tag is located

**Returns:** Stub or proxy

**Throws:**
- `Exception` - In case of error during processing

**setINSByte(byte)**

```java
public void setINSByte(byte ins)
```

Sets the ISO 7816-4 header INS byte to use for Java Card RMI method invocation commands.

**Parameters:**
- `ins` - the ISO 7816-4 INS command header byte to use for Java Card RMI invocation commands

**getINSByte()**

```java
public byte getINSByte()
```
getRemoteRefFormat() returns the configured ISO 7816-4 command INS byte to be used in the Java Card remote method invocation command

**Returns:** the ISO-7816-4 INS byte

**Since:** Java Card 2.2

```java
public abstract byte getRemoteRefFormat()
```

returns the format of JCRMI remote object reference descriptor supported. It returns one of:

- REF_FORMAT_NONE
- REF_FORMAT_CLASS
- REF_FORMAT_INTERFACES

**Returns:** one of the REF_FORMAT.. values defined above
CardRuntimeExceptionSubclass com.sun.javacard.javax.smartcard.rmiclient
getRemoteRefFormat()

com.sun.javacard.javax.smartcard.rmiclient
CardRuntimeExceptionSubclass

Declaration
public class CardRuntimeExceptionSubclass extends CardRuntimeException

java.lang.Object
   |---java.lang.Throwable
      |   |---java.lang.Exception
      |     |---java.lang.RuntimeException
      |        |---javacard.framework.CardRuntimeException
                      |---com.sun.javacard.javax.smartcard.rmiclient.

All Implemented Interfaces: java.io.Serializable

Description
This exception class represents a subclass of CardRuntimeException on the card. The CardRuntimeException class on the card defines a field reason and two accessor methods getReason() and setReason(). The reason field encapsulates exception cause identifier in Java Card. All Java Card unchecked Exception classes on the card should extend CardRuntimeException.

Member Summary

<table>
<thead>
<tr>
<th>Constructors</th>
</tr>
</thead>
<tbody>
<tr>
<td>CardRuntimeExceptionSubclass(java.lang.String msg, short reason)</td>
</tr>
<tr>
<td>Construct a CardRuntimeExceptionSubclass instance with the specified reason and the specified error message.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.lang.String getMessage()</td>
</tr>
<tr>
<td>Returns the error message string of this throwable object.</td>
</tr>
</tbody>
</table>

Inherited Member Summary

Methods inherited from interface CardRuntimeException
getReason(), setReason(short), throwIt(short)

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
Inherited Member Summary

Methods inherited from class java.lang.Throwable

fillInStackTrace, getLocalizedMessage, printStackTrace, printStackTrace, printStackTrace, toString

Constructors

CardRuntimeExceptionSubclass(String, short)

public CardRuntimeExceptionSubclass(java.lang.String msg, short reason)

Construct a CardRuntimeExceptionSubclass instance with the specified reason and the specified error message.

Parameters:

msg - the associated message string
reason - the reason for the exception

Methods

getMessage()

public java.lang.String getMessage()

Returns the error message string of this throwable object.

Overrides: getMessage in class Throwable

Returns: the error message string of this Throwable object if it was created with an error message string; or null if it was created with no error message.
Declaration

public class CryptoExceptionSubclass extends CryptoException

java.lang.Object
   |--java.lang.Throwable
      |--java.lang.Exception
         |--java.lang.RuntimeException
            |--javacard.framework.CardRuntimeException
               |--javacard.security.CryptoException
                  |--com.sun.javacard.javax.smartcard.rmiclient.

CryptoExceptionSubclass

All Implemented Interfaces: java.io.Serializable

Description
This exception class represents a card subclass of CryptoException. CryptoException represents a cryptography-related exception.

Member Summary

<table>
<thead>
<tr>
<th>Constructors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Inherited Member Summary

Fields inherited from interface CryptoException

ILLEGAL_USE, ILLEGAL_VALUE, INVALID_INIT, NO_SUCH_ALGORITHM, UNINITIALIZED_KEY

Methods inherited from interface CardRuntimeException

getReason(), setReason(short)

Methods inherited from interface CryptoException
Inherited Member Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>throwIt(short)</td>
</tr>
</tbody>
</table>

Constructors

CryptoExceptionSubclass(String, short)

```java
public CryptoExceptionSubclass(java.lang.String msg, short reason)
```

Constructs an CryptoExceptionSubclass with the specified reason and specified error message.

**Parameters:**
- `msg`: the associated message string
- `reason`: the reason for the exception.

Methods

getMessage()

```java
public java.lang.String getMessage()
```

Returns the error message string of this throwable object.

**Overrides:** getMessage in class Throwable

**Returns:** the error message string of this Throwable object if it was created with an error message string; or null if it was created with no error message.
Declaration

public class ISOExceptionSubclass extends ISOException

java.lang.Object
  |---java.lang.Throwable
  |   |---java.lang.Exception
  |     |---java.lang.RuntimeException
  |       |---javacard.framework.CardRuntimeException
  |         |---javacard.framework.ISOException
  |           |---com.sun.javacard.javax.smartcard.rmiclient.

All Implemented Interfaces: java.io.Serializable

Description

This exception class represents a card subclass of ISOException. ISOException class encapsulates an ISO 7816-4 response status word as its reason code.

Member Summary

<table>
<thead>
<tr>
<th>Constructors</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISOExceptionSubclass(java.lang.String msg, short reason)</td>
</tr>
<tr>
<td>Constructs an ISOExceptionSubclass with the specified reason and specified error message.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.lang.String getMessage()</td>
</tr>
<tr>
<td>Returns the error message string of this throwable object.</td>
</tr>
</tbody>
</table>

Inherited Member Summary

Methods inherited from interface CardRuntimeException

getReason(), setReason(short)

Methods inherited from interface ISOException

throwIt(short)

Methods inherited from class java.lang.Object
ISOExceptionSubclass

ISOExceptionSubclass(String, short)

Constructors

ISOExceptionSubclass(String, short)

public ISOExceptionSubclass(java.lang.String msg, short reason)

Constructs an ISOExceptionSubclass with the specified reason and specified error message.

Parameters:

msg - the associated message string

reason - the ISO 7816-4 defined status word

Methods

getMessage()

public java.lang.String getMessage()

Returns the error message string of this throwable object.

Overrides: getMessage in class Throwable

Returns: the error message string of this Throwable object if it was created with an error message string; or null if it was created with no error message.
Declaration

public class JavaCardRMIConnect

java.lang.Object
   | +-- com.sun.javacard.javax.smartcard.rmiClient.JavaCardRMIConnect

Description

The JavaCardRMIConnect class is used by the client application to initialize a Java Card RMI session and obtain an initial remote reference. The following code shows how a typical client application which uses an underlying OCF framework interacts with this class.

```java
...  
CardRequest cr = new CardRequest( 5 ); //wait 5 seconds  
cr.setWaitBehavior( CardRequest.ANYCARD );  
SmartCard myCard = SmartCard.waitForCard( cr, null, JCRMICardService.class);  
CardAccessor myCS = (CardAccessor) myCard.getCardService( OCFCardAccessor.  
class, true );  
// create and initialize JavaCard RMI session  
JavaCardRMIConnect jcRMI = new JavaCardConnect( myCS );  
byte[] appAID = new byte[] { 0x1,0x2,0x3,0x4,0x5,0x6, 0x7,0x8 };  
jcRMI.selectApplet( appAID );  
initInterface iR = (initInterface) jcRMI.getInitialReference();  
...  
JPInterface jpi = iR.getJPInterface();  
short balance = jpi.getBalance();
```

Member Summary

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>JavaCardRMIConnect(CardAccessor cA)</strong></td>
</tr>
<tr>
<td>Creates a new JavaCardRMIConnect instance using the specified</td>
</tr>
<tr>
<td>CardAccessor</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getInitialReference()</strong></td>
</tr>
<tr>
<td>Returns the initial remote object for the selected Java Card applet.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>selectApplet(byte[] appletAID)</strong></td>
</tr>
<tr>
<td>Selects the specified Java Card Applet</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>selectApplet(byte[] appletAID, CardObjectFactory cOF)</strong></td>
</tr>
<tr>
<td>Selects the specified Java Card Applet and specifies a CardObjectFactory to use for the Java Card RMI stub generation</td>
</tr>
</tbody>
</table>

Inherited Member Summary

Methods inherited from class java.lang.Object
Constructors

**JavaCardRMIConnect(CardAccessor)**

public `JavaCardRMIConnect(com.sun.javacard.javax.smartcard.rmiclient.CardAccessor cA)`

Creates a new `JavaCardRMIConnect` instance using the specified `CardAccessor`

**Parameters:**
- `cA` - the `CardAccessor` instance to use for card communication

Methods

**selectApplet(byte[])**

public byte[] `selectApplet(byte[] appletAID)` throws IOException

Selects the specified Java Card Applet

**Parameters:**
- `appletAID` - the AID of the Java Card applet

**Returns:** the response received from the applet selection command. The response consists of 2 bytes of status (SW1, SW2) followed by the response data. The response data is interpreted in an FCI format to configure the Java Card RMI session.

**Throws:**
- `IOException` - if a card communication error occurs

**selectApplet(byte[], CardObjectFactory)**

public byte[] `selectApplet(byte[] appletAID, com.sun.javacard.javax.smartcard.rmiclient.CardObjectFactory cOF)` throws IOException

Selects the specified Java Card Applet and specifies a CardObjectFactory to use for the Java Card RMI stub generation

**Parameters:**
- `appletAID` - the AID of the Java Card applet
- `cOF` - the card stub factory instance to use for Java Card RMI stub/proxy instantiation.

**Returns:** the response received from the applet selection command. The response consists of 2 bytes of status (SW1, SW2) followed by the response data. The response data is interpreted in an FCI format to configure the Java Card RMI session.

**Throws:**
- `IOException` - if a card communication error occurs
getInitialReference()

```java
public java.rmi.Remote getInitialReference()
    throws RemoteException
```

Returns the initial remote object for the selected Java Card applet. The initial remote object reference was returned during the applet selection.

**Returns:** the Java Card applet’s initial remote object

**Throws:**

RemoteException - if a Java Card RMI protocol error occurs
com.sun.javacard.javax.smartcard.rmiclient

PINExceptionSubclass

Declaration
public class PINExceptionSubclass extends PINException

description
This exception class represents a card subclass of PINException. PINException represents an OwnerPIN class access-related exception.

Member Summary

Constructors
PINExceptionSubclass(java.lang.String msg, short reason)
Constructs a PINExceptionSubclass with the specified reason and specified error message.

Methods
java.lang.String getMessage()
Returns the error message string of this throwable object.

Inherited Member Summary

Fields inherited from interface PINException
ILLEGAL_VALUE

Methods inherited from interface CardRuntimeException
getReason(), setReason(short)

Methods inherited from class java.lang.Object
PINExceptionSubclass(String, short)

Constructors

PINExceptionSubclass(String, short)

public PINExceptionSubclass(java.lang.String msg, short reason)

Constructs an PINExceptionSubclass with the specified reason and specified error message.

Parameters:
- msg - the associated message string
- reason - the reason for the exception.

Methods

getMessage()

public java.lang.String getMessage()

Returns the error message string of this throwable object.

Overrides: getMessage in class Throwable

Returns: the error message string of this Throwable object if it was created with an error message string; or null if it was created with no error message.
Declaration

public class ServiceExceptionSubclass extends ServiceException

java.lang.Object
    |---java.lang.Throwable
        |---java.lang.Exception
            |---java.lang.RuntimeException
                |---javacard.framework.CardRuntimeException
                    |---javacard.framework.service.ServiceException
                        |---com.sun.javacard.javax.smartcard.rmiclient.

All Implemented Interfaces: java.io.Serializable

Description

This exception class represents a card subclass of ServiceException. ServiceException represents a service framework related exception.

Member Summary

Constructors

| ServiceExceptionSubclass(java.lang.String msg, short reason) |
| Constructs an ServiceExceptionSubclass with the specified reason and specified error message. |

Methods

| java.lang.String getMessage() |
| Returns the error message string of this throwable object. |

Inherited Member Summary

Fields inherited from interface ServiceException

CANNOT_ACCESS_IN_COMMAND, CANNOT_ACCESS_OUT_COMMAND, COMMAND_DATA_TOO_LONG, COMMAND_IS_FINISHED, DISPATCH_TABLE_FULL, ILLEGAL_PARAM, REMOTE_OBJECT_NOTEXPORTED

Methods inherited from interface CardRuntimeException

getAddress(), setReason(short)
### Inherited Member Summary

**Methods inherited from class java.lang.Object**
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

**Methods inherited from interface ServiceException**
- throwIt(short)

**Methods inherited from class java.lang.Throwable**
- fillInStackTrace, getLocalizedMessage, printStackTrace, printStackTrace, printStackTrace, toString

---

### Constructors

**ServiceExceptionSubclass(String, short)**

```java
public ServiceExceptionSubclass(java.lang.String msg, short reason)
```

Constructs an ServiceExceptionSubclass with the specified reason and specified error message.

**Parameters:**
- `msg` - the associated message string
- `reason` - the reason for the exception.

---

### Methods

**getMessage()**

```java
public java.lang.String getMessage()
```

Returns the error message string of this throwable object.

**Overrides:** getMessage in class Throwable

**Returns:** the error message string of this Throwable object if it was created with an error message string; or null if it was created with no error message.
com.sun.javacard.javax.smartcard.rmiclient

SystemExceptionSubclass

Declaration
public class SystemExceptionSubclass extends SystemException

java.lang.Object
  +--java.lang.Throwable
    +--java.lang.Exception
      +--java.lang.RuntimeException
        +--javacard.framework.CardRuntimeException
          +--javacard.framework.SystemException
            +--com.sun.javacard.javax.smartcard.rmiclient.

All Implemented Interfaces: java.io.Serializable

Description
This exception class represents a card subclass of SystemException. SystemException represents a JCSystem class related exception.

Member Summary

Constructors

SystemExceptionSubclass(java.lang.String msg, short reason)
Constructs an SystemExceptionSubclass with the specified reason and specified error message.

Methods

java.lang.String getMessage()
Returns the error message string of this throwable object.

Inherited Member Summary

Fields inherited from interface SystemException
ILLEGAL_AID, ILLEGAL_TRANSIENT, ILLEGAL_USE, ILLEGAL_VALUE, NO_RESOURCE, NO_TRANSIENT_SPACE

Methods inherited from interface CardRuntimeException
getReason(), setReason(short)
Constructors

SystemExceptionSubclass(String, short)

public SystemExceptionSubclass(java.lang.String msg, short reason)

Constructs an SystemExceptionSubclass with the specified reason and specified error message.

Parameters:

msg - the associated message string
reason - the reason for the exception.

Methods

getMessage()

public java.lang.String getMessage()

Returns the error message string of this throwable object.

Overrides: getMessage in class Throwable

Returns: the error message string of this Throwable object if it was created with an error message string; or null if it was created with no error message.
com.sun.javacard.javax.smartcard.rmiclient

**TransactionExceptionSubclass**

### Declaration

public class **TransactionExceptionSubclass** extends TransactionException

java.lang.Object
- java.lang.Throwable
  - java.lang.Exception
    - java.lang.RuntimeException
      - javacard.framework.CardRuntimeException
        - javacard.framework.TransactionException
          - com.sun.javacard.javax.smartcard.rmiclient.

**TransactionExceptionSubclass**

### All Implemented Interfaces:

java.io.Serializable

### Description

This exception class represents a card subclass of TransactionException. TransactionException represents an exception in the transaction subsystem.

### Member Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>getMessage()</td>
</tr>
</tbody>
</table>

- **getMessage()**
  Returns the error message string of this throwable object.

### Inherited Member Summary

- **getReason()**, **setReason(short)**
  
- **BUFFER_FULL**, **INTERNAL_FAILURE**, **IN_PROGRESS**, **NOT_IN_PROGRESS**
  
- **getReason()**, **setReason(short)**
### Inherited Member Summary

**Methods inherited from class java.lang.Object**
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

**Methods inherited from class java.lang.Throwable**
- fillInStackTrace, getLocalizedMessage, printStackTrace, printStackTrace, toString

**Methods inherited from interface TransactionException**
- throwIt(short)

## Constructors

**TransactionExceptionSubclass(String, short)**

```java
public TransactionExceptionSubclass(java.lang.String msg, short reason)
```

Constructs a TransactionExceptionSubclass with the specified reason and specified error message.

**Parameters:**
- `msg` - the associated message string
- `reason` - the reason for the exception.

## Methods

**getMessage()**

```java
public java.lang.String getMessage()
```

Returns the error message string of this throwable object.

**Overrides:** getMessage in class Throwable

**Returns:** the error message string of this Throwable object if it was created with an error message string; or null if it was created with no error message.
UserExceptionSubclass

Declaration

```java
public class UserExceptionSubclass extends UserException
```

```java
java.lang.Object
 | +--java.lang.Throwable
 |    | +--java.lang.Exception
 |    |    +--javacard.framework.CardException
 |    |    | +--javacard.framework.UserException
 |    +--com.sun.javacard.javax.smartcard.rmiclient.
```

All Implemented Interfaces: java.io.Serializable

Description

This exception class represents a card subclass of UserException. UserException represents a User exception.

Member Summary

<table>
<thead>
<tr>
<th>Constructors</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserExceptionSubclass(java.lang.String msg, short reason)</td>
</tr>
<tr>
<td>Constructs an UserExceptionSubclass with the specified reason and specified error message.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.lang.String getMessage()</td>
</tr>
<tr>
<td>Returns the error message string of this throwable object.</td>
</tr>
</tbody>
</table>

Inherited Member Summary

Methods inherited from interface CardException

getReason(), setReason(short)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Methods inherited from class java.lang.Throwable

fillInStackTrace, getLocalizedMessage, printStackTrace, printStackTrace, printStackTrace, toString
UserExceptionSubclass com.sun.javacard.javax.smartcard.rmiclient
UserExceptionSubclass(String, short)

### Inherited Member Summary

Methods inherited from interface UserException

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>throwIt(short)</td>
</tr>
</tbody>
</table>

### Constructors

**UserExceptionSubclass(String, short)**

```java
public UserExceptionSubclass(java.lang.String msg, short reason)
```

Constructs an UserExceptionSubclass with the specified reason and specified error message.

**Parameters:**

- `msg` - the associated message string
- `reason` - the reason for the exception.

### Methods

**getMessage()**

```java
public java.lang.String getMessage()
```

Returns the error message string of this throwable object.

**Overrides:** getMessage in class Throwable

**Returns:** the error message string of this Throwable object if it was created with an error message string; or null if it was created with no error message.
Package
javacard.framework

Description
Provides Java Card exceptions that may be re-thrown on the client.

<table>
<thead>
<tr>
<th>Class Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exceptions</td>
</tr>
<tr>
<td>APDUException</td>
</tr>
<tr>
<td>CardException</td>
</tr>
<tr>
<td>CardRuntimeException</td>
</tr>
<tr>
<td>ISOException</td>
</tr>
<tr>
<td>PINException</td>
</tr>
<tr>
<td>SystemException</td>
</tr>
<tr>
<td>TransactionException</td>
</tr>
<tr>
<td>UserException</td>
</tr>
</tbody>
</table>
APDUException

javacard.framework

APDUException

Declaration

public class APDUException extends CardRuntimeException

java.lang.Object
  +--java.lang.Throwables
    +--java.lang.Throwable
      +--java.lang.Exception
        +--java.lang.RuntimeException
          +--javacard.framework.CardRuntimeException
            +--javacard.framework.APDUException

All Implemented Interfaces: java.io.Serializable

Direct Known Subclasses: APDUExceptionSubclass

Description

APDUException represents an APDU related exception.

Member Summary

<table>
<thead>
<tr>
<th>Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>static short BAD_LENGTH</td>
</tr>
<tr>
<td>static short BUFFER_BOUNDS</td>
</tr>
<tr>
<td>static short ILLEGAL_USE</td>
</tr>
<tr>
<td>static short IO_ERROR</td>
</tr>
<tr>
<td>static short NO_T0_GETRESPONSE</td>
</tr>
<tr>
<td>static short NO_T0_REISSUE</td>
</tr>
</tbody>
</table>

BAD_LENGTH
This reason code is used by the APDU.setOutgoingLength() method to indicate that the length parameter is greater than 256 or if non BLOCK CHAINED data transfer is requested and len is greater than (IFSD-2), where IFSD is the Outgoing Block Size.

BUFFER_BOUNDS
This reason code is used by the APDU.sendBytes() method to indicate that the sum of buffer offset parameter and the byte length parameter exceeds the APDU buffer size.

ILLEGAL_USE
This APDUException reason code indicates that the method should not be invoked based on the current state of the APDU.

IO_ERROR
This reason code indicates that an unrecoverable error occurred in the I/O transmission layer.

NO_T0_GETRESPONSE
This reason code indicates that during T=0 protocol, the CAD did not return a GET RESPONSE command in response to a <61xx> response status to send additional data.

NO_T0_REISSUE
This reason code indicates that during T=0 protocol, the CAD did not reissue the same APDU command with the corrected length in response to a <6Cxx> response status to request command reissue with the specified length.
### Member Summary

<table>
<thead>
<tr>
<th>static short</th>
<th>T1_IFD_ABORT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This reason code indicates that during T=1 protocol, the CAD returned an ABORT S-Block command and aborted the data transfer.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Constructors</th>
</tr>
</thead>
<tbody>
<tr>
<td>APDUException(short reason)</td>
</tr>
<tr>
<td>Constructs an APDUException.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>static void throwIt(short reason)</td>
</tr>
<tr>
<td>Throws an instance of APDUException with the specified reason.</td>
</tr>
</tbody>
</table>

### Inherited Member Summary

Methods inherited from interface CardRuntimeException
getReason(), setReason(short)

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Methods inherited from class java.lang.Throwable
fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString

### Fields

**ILLEGAL_USE**

public static final short ILLEGAL_USE

This APDUException reason code indicates that the method should not be invoked based on the current state of the APDU.

**BUFFER_BOUNDS**

public static final short BUFFER_BOUNDS

This reason code is used by the APDU.sendBytes() method to indicate that the sum of buffer offset parameter and the byte length parameter exceeds the APDU buffer size.

**BAD_LENGTH**

public static final short BAD_LENGTH

This reason code is used by the APDU.setOutgoingLength() method to indicate that the length parameter is greater than 256 or if non BLOCK CHAINED data transfer is requested and len is greater than (IFSD-2), where IFSD is the Outgoing Block Size.
APDUException javacard.framework

IO_ERROR

public static final short IO_ERROR
This reason code indicates that an unrecoverable error occurred in the I/O transmission layer.

NO_T0_GETRESPONSE

public static final short NO_T0_GETRESPONSE
This reason code indicates that during T=0 protocol, the CAD did not return a GET RESPONSE command in response to a <61xx> response status to send additional data. The outgoing transfer has been aborted. No more data or status can be sent to the CAD in this APDU.process() method.

T1_IFD_ABORT

public static final short T1_IFD_ABORT
This reason code indicates that during T=1 protocol, the CAD returned an ABORT S-Block command and aborted the data transfer. The incoming or outgoing transfer has been aborted. No more data can be received from the CAD. No more data or status can be sent to the CAD in this APDU.process() method.

NO_T0_REISSUE

public static final short NO_T0_REISSUE
This reason code indicates that during T=0 protocol, the CAD did not reissue the same APDU command with the corrected length in response to a <6Cxx> response status to request command reissue with the specified length. The outgoing transfer has been aborted. No more data or status can be sent to the CAD in this APDU.process() method.

Constructor

APDUException(short)

public APDUException(short reason)
Constructs an APDUException.

Parameters:
reason - the reason for the exception.

Methods

throwIt(short)

public static void throwIt(short reason)
Throws an instance of APDUException with the specified reason.

Parameters:
reason - the reason for the exception.

Throws:
APDUException - always.
javacard.framework

CardException

Declaration
public class CardException extends java.lang.Exception

java.lang.Object
  |---java.lang.Throwable
    |---java.lang.Exception
        |---javacard.framework.CardException

All Implemented Interfaces: java.io.Serializable

Direct Known Subclasses: CardExceptionSubclass, UserException

Description
The CardException class defines a field reason and two accessor methods getReason() and setReason(). The reason field encapsulates exception cause identifier in Java Card. All Java Card checked Exception classes should extend CardException.

Member Summary

<table>
<thead>
<tr>
<th>Constructors</th>
</tr>
</thead>
<tbody>
<tr>
<td>CardException(short reason)</td>
</tr>
<tr>
<td>Construct a CardException instance with the specified reason.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>short getReason()</td>
</tr>
<tr>
<td>Get reason code</td>
</tr>
<tr>
<td>void setReason(short reason)</td>
</tr>
<tr>
<td>Set reason code.</td>
</tr>
<tr>
<td>static void throwIt(short reason)</td>
</tr>
<tr>
<td>Throw an instance of CardException class with the specified reason.</td>
</tr>
</tbody>
</table>

Inherited Member Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Methods inherited from class java.lang.Throwable
fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, toString
CardException

CardException(short)

    public CardException(short reason)
    Construct a CardException instance with the specified reason.
    Parameters:  
      reason - the reason for the exception

Methods

getReason()

    public short getReason()
    Get reason code
    Returns:  the reason for the exception

setReason(short)

    public void setReason(short reason)
    Set reason code.
    Parameters:  
      reason - the reason for the exception

throwIt(short)

    public static void throwIt(short reason)
    throws CardException
    Throw an instance of CardException class with the specified reason.
    Parameters:  
      reason - the reason for the exception
    Throws:  
      CardException - always.
javacard.framework

CardRuntimeException

Declaration

public class CardRuntimeException extends java.lang.RuntimeException

java.lang.Object
  |---java.lang.Throwable
  |   |---java.lang.Exception
  |   |   |---java.lang.RuntimeException
  |   |---javacard.framework.CardRuntimeException

All Implemented Interfaces: java.io.Serializable

Direct Known Subclasses: APDUException, CardRuntimeExceptionSubclass, CryptoException, ISOException, PINException, ServiceException, SystemException, TransactionException

Description

The CardRuntimeException class defines a field reason and two accessor methods getReason() and setReason(). The reason field encapsulates exception cause identifier in Java Card. All Java Card unchecked Exception classes should extend CardRuntimeException.

Member Summary

<table>
<thead>
<tr>
<th>Constructors</th>
</tr>
</thead>
<tbody>
<tr>
<td>CardRuntimeException(short reason)</td>
</tr>
<tr>
<td>Construct a CardRuntimeException instance with the specified reason.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>short getReason()</td>
</tr>
<tr>
<td>Get reason code</td>
</tr>
<tr>
<td>void setReason(short reason)</td>
</tr>
<tr>
<td>Set reason code.</td>
</tr>
<tr>
<td>static void throwIt(short reason)</td>
</tr>
<tr>
<td>Throw an instance of the CardRuntimeException class with the specified reason.</td>
</tr>
</tbody>
</table>

Inherited Member Summary

Methods inherited from class java.lang.Object
CardRuntimeException

CardRuntimeException(short)

Constructors

public CardRuntimeException(short reason)

Construct a CardRuntimeException instance with the specified reason.

Parameters:
reason - the reason for the exception

Methods

getReason()

public short getReason()

Get reason code

Returns: the reason for the exception

setReason(short)

public void setReason(short reason)

Set reason code.

Parameters:
reason - the reason for the exception

throwIt(short)

public static void throwIt(short reason)

throws CardRuntimeException

Throw an instance of the CardRuntimeException class with the specified reason.

Parameters:
reason - the reason for the exception

Throws:
CardRuntimeException - always.
javacard.framework

ISOException

Declaration
public class ISOException extends CardRuntimeException

java.lang.Object
    |-- java.lang.Throwable
        |-- java.lang.Exception
            |-- java.lang.RuntimeException
                |-- javacard.framework.CardRuntimeException
                    |-- javacard.framework.ISOException

All Implemented Interfaces: java.io.Serializable

Direct Known Subclasses: ISOExceptionSubclass

Description
ISOException class encapsulates an ISO 7816-4 response status word as its reason code.

<table>
<thead>
<tr>
<th>Member Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructors</td>
</tr>
<tr>
<td>ISOException(short sw)</td>
</tr>
<tr>
<td>Constructs an ISOException instance with the specified status word.</td>
</tr>
<tr>
<td>Methods</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>throwIt(short sw)</td>
</tr>
<tr>
<td>Throws an instance of the ISOException class with the specified status word.</td>
</tr>
</tbody>
</table>

Inherited Member Summary

Methods inherited from interface CardRuntimeException
getReason(), setReason(short)

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

Methods inherited from class java.lang.Throwable
Constructors

ISOException(short)

public ISOException(short sw)
Constructs an ISOException instance with the specified status word.

Parameters:
sw - the ISO 7816-4 defined status word

Methods

throwIt(short)

public static void throwIt(short sw)
Throws an instance of the ISOException class with the specified status word.

Parameters:
sw - ISO 7816-4 defined status word

Throws:
ISOException - always.
javacard.framework

PINException

Declaration
public class PINException extends CardRuntimeException

java.lang.Object
    |-- java.lang.Throwable
    |    |-- java.lang.Exception
    |    |    |-- java.lang.RuntimeException
    |    |    |    |-- javacard.framework.CardRuntimeException
    |    |    |    |    |-- javacard.framework.PINException

All Implemented Interfaces: java.io.Serializable

Direct Known Subclasses: PINExceptionSubclass

Description
PINException represents a OwnerPIN class access-related exception.

Member Summary

<table>
<thead>
<tr>
<th>Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>static short ILLEGAL_VALUE</td>
</tr>
<tr>
<td>This reason code is used to indicate that one or more input parameters is out of allowed bounds.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Constructors</th>
</tr>
</thead>
<tbody>
<tr>
<td>PINException(short reason)</td>
</tr>
<tr>
<td>Constructs a PINException.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>static void throwIt(short reason)</td>
</tr>
<tr>
<td>Throws an instance of PINException with the specified reason.</td>
</tr>
</tbody>
</table>

Inherited Member Summary

| Methods inherited from interface CardRuntimeException |
| getReason(), setReason(short) |

| Methods inherited from class java.lang.Object |
**Fields**

**ILLEGAL_VALUE**

public static final short **ILLEGAL_VALUE**

This reason code is used to indicate that one or more input parameters is out of allowed bounds.

**Constructors**

**PINException(short)**

public **PINException**(short reason)

Constructs a PINException. To conserve on resources use **throwIt()** to use the JCRE owned instance of this class.

**Parameters:**

  reason - the reason for the exception.

**Methods**

**throwIt(short)**

public static void **throwIt**(short reason)

Throws an instance of **PINException** with the specified reason.

**Parameters:**

  reason - the reason for the exception.

**Throws:**

  **PINException** - always.
javacard.framework
SystemException

Declaration
public class SystemException extends CardRuntimeException

java.lang.Object
    +--java.lang.Throwable
        +--java.lang.Exception
            +--java.lang.RuntimeException
                +-- javacard.framework.CardRuntimeException
                    +--javacard.framework.SystemException

All Implemented Interfaces: java.io.Serializable

Direct Known Subclasses: SystemExceptionSubclass

Description
SystemException represents a JCSystem class related exception.

Member Summary

Fields
<table>
<thead>
<tr>
<th>static short</th>
<th>ILLEGAL_AID</th>
</tr>
</thead>
</table>
|              | This reason code is used by the javacard.framework.Applet.register() method to indicate that the input AID parameter is not a legal AID value.

<table>
<thead>
<tr>
<th>static short</th>
<th>ILLEGAL_TRANSIENT</th>
</tr>
</thead>
</table>
|              | This reason code is used to indicate that the request to create a transient object is not allowed in the current applet context.

<table>
<thead>
<tr>
<th>static short</th>
<th>ILLEGAL_USE</th>
</tr>
</thead>
</table>
|              | This reason code is used to indicate that the requested function is not allowed.

<table>
<thead>
<tr>
<th>static short</th>
<th>ILLEGAL_VALUE</th>
</tr>
</thead>
</table>
|              | This reason code is used to indicate that one or more input parameters is out of allowed bounds.

<table>
<thead>
<tr>
<th>static short</th>
<th>NO_RESOURCE</th>
</tr>
</thead>
</table>
|              | This reason code is used to indicate that there is insufficient resource in the Card for the request.

<table>
<thead>
<tr>
<th>static short</th>
<th>NO_TRANSIENT_SPACE</th>
</tr>
</thead>
</table>
|              | This reason code is used by the makeTransient..() methods to indicate that no room is available in volatile memory for the requested object.

Constructors
SystemException(short reason)
    Constructs a SystemException.

Methods
### Member Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static void throwIt(short reason)</td>
<td>Throws an instance of SystemException with the specified reason.</td>
</tr>
</tbody>
</table>

### Inherited Member Summary

Methods inherited from interface CardRuntimeException

- getReason(), setReason(short)

Methods inherited from class java.lang.Object

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

Methods inherited from class java.lang.Throwable

- fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, toString

### Fields

**ILLEGAL_VALUE**

- public static final short ILLEGAL_VALUE

  This reason code is used to indicate that one or more input parameters is out of allowed bounds.

**NO_TRANSIENT_SPACE**

- public static final short NO_TRANSIENT_SPACE

  This reason code is used by the makeTransient..() methods to indicate that no room is available in volatile memory for the requested object.

**ILLEGAL_TRANSIENT**

- public static final short ILLEGAL_TRANSIENT

  This reason code is used to indicate that the request to create a transient object is not allowed in the current applet context. See Java Card Runtime Environment (JCRE) Specification, section 6.2.1 for details.

**ILLEGAL_AID**

- public static final short ILLEGAL_AID

  This reason code is used by the javacard.framework.Applet.register() method to indicate that the input AID parameter is not a legal AID value.

**NO_RESOURCE**

- public static final short NO_RESOURCE

  This reason code is used to indicate that there is insufficient resource in the Card for the request.
For example, the Java Card Virtual Machine may throw this exception reason when there is insufficient heap space to create a new instance.

**ILLEGAL_USE**

    public static final short ILLEGAL_USE

This reason code is used to indicate that the requested function is not allowed. For example, `JCSystem.requestObjectDeletion()` method throws this exception if the object deletion mechanism is not implemented.

---

### Constructors

**SystemException(short)**

    public SystemException(short reason)

Constructs a SystemException.

**Parameters:**

- `reason` - the reason for the exception.

---

### Methods

**throwIt(short)**

    public static void throwIt(short reason)

Throws an instance of `SystemException` with the specified reason.

**Parameters:**

- `reason` - the reason for the exception.

** Throws:**

- `SystemException` - always.
TransactionException

Declaration
public class TransactionException extends CardRuntimeException

java.lang.Object
   \+- java.lang.Throwable
      \+- java.lang.Exception
         \+- java.lang.RuntimeException
            \+- javacard.framework.CardRuntimeException
               \+- javacard.framework.TransactionException

All Implemented Interfaces: java.io.Serializable

Direct Known Subclasses: TransactionExceptionSubclass

Description
TransactionException represents an exception in the transaction subsystem.

<table>
<thead>
<tr>
<th>Member Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fields</strong></td>
</tr>
<tr>
<td>static short</td>
</tr>
<tr>
<td>static short</td>
</tr>
<tr>
<td>static short</td>
</tr>
<tr>
<td>static short</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Constructors</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>TransactionException(short reason) Constructs a TransactionException with the specified reason.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Methods</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>static void throwIt(short reason) Throws an instance of TransactionException with the specified reason.</td>
</tr>
</tbody>
</table>
Inherited Member Summary

Methods inherited from interface CardRuntimeException
getReason(), setReason(short)

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Methods inherited from class java.lang.Throwable
fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString

Fields

IN_PROGRESS

public static final short IN_PROGRESS
This reason code is used by the beginTransaction method to indicate a transaction is already in progress.

NOT_IN_PROGRESS

public static final short NOT_IN_PROGRESS
This reason code is used by the abortTransaction and commitTransaction methods when a transaction is not in progress.

BUFFER_FULL

public static final short BUFFER_FULL
This reason code is used during a transaction to indicate that the commit buffer is full.

INTERNAL_FAILURE

public static final short INTERNAL_FAILURE
This reason code is used during a transaction to indicate an internal JCRE problem (fatal error).

Constructors

TransactionException(short)

public TransactionException(short reason)
Constructs a TransactionException with the specified reason.
Methods

throwIt(short)

public static void throwIt(short reason)

Throws an instance of TransactionException with the specified reason.

Throws:

TransactionException - always.
javacard.framework

UserException

Declaration
public class UserException extends CardException

java.lang.Object
   |---java.lang.Throwable
      |   |---java.lang.Exception
         |   +--javacard.framework.CardException
             |   +--javacard.framework.UserException

All Implemented Interfaces: java.io.Serializable

Direct Known Subclasses: UserExceptionSubclass

Description
UserException represents a User exception.

Member Summary

Constructors

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserException()</td>
<td>Constructs a UserException with reason = 0.</td>
</tr>
<tr>
<td>UserException(short reason)</td>
<td>Constructs a UserException with the specified reason.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static void throwIt(short reason)</td>
<td>Throws an instance of UserException with the specified reason.</td>
</tr>
</tbody>
</table>

Inherited Member Summary

Methods inherited from interface CardException
getReason(), setReason(short)

Methods inherited from class java.lang.Object
clon, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Methods inherited from class java.lang.Throwable
**UserException**

**UserException()**

```java
public UserException()

Constructs a UserException with reason = 0.
```

**UserException(short)**

```java
public UserException(short reason)

Constructs a UserException with the specified reason.

Parameters:
- reason - the reason for the exception.
```

**Methods**

**throwIt(short)**

```java
public static void throwIt(short reason)

throws UserException

Throws an instance of UserException with the specified reason.

Parameters:
- reason - the reason for the exception.

Throws:
- UserException - always.
```
Package
javacard.framework.service

Description
Provides Java Card exceptions that may be re-thrown on the client.

<table>
<thead>
<tr>
<th>Class Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exceptions</td>
</tr>
<tr>
<td><strong>ServiceException</strong></td>
</tr>
</tbody>
</table>
Declaration
public class ServiceException extends CardRuntimeException

java.lang.Object
   +--java.lang.Throwable
      +--java.lang.Exception
         +--java.lang.RuntimeException
            +--javacard.framework.CardRuntimeException
               +--javacard.framework.service.ServiceException

All Implemented Interfaces: java.io.Serializable

Direct Known Subclasses: ServiceExceptionSubclass

Description
ServiceException represents a service framework related exception.

Member Summary

<table>
<thead>
<tr>
<th>Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>static short CANNOT_ACCESS_IN_COMMAND</td>
</tr>
<tr>
<td>This reason code is used to indicate that the command in the APDU object cannot be accessed for input processing.</td>
</tr>
<tr>
<td>static short CANNOT_ACCESS_OUT_COMMAND</td>
</tr>
<tr>
<td>This reason code is used to indicate that the command in the APDU object cannot be accessed for output processing.</td>
</tr>
<tr>
<td>static short COMMAND_DATA_TOO_LONG</td>
</tr>
<tr>
<td>This reason code is used to indicate that the incoming data for a command in the APDU object does not fit in the APDU buffer.</td>
</tr>
<tr>
<td>static short COMMAND_IS_FINISHED</td>
</tr>
<tr>
<td>This reason code is used to indicate that the command in the APDU object has been completely processed.</td>
</tr>
<tr>
<td>static short DISPATCH_TABLE_FULL</td>
</tr>
<tr>
<td>This reason code is used to indicate that a dispatch table is full</td>
</tr>
<tr>
<td>static short ILLEGAL_PARAM</td>
</tr>
<tr>
<td>This reason code is used to indicate that an input parameter is not allowed.</td>
</tr>
<tr>
<td>static short REMOTE_OBJECT_NOT_EXPORTED</td>
</tr>
<tr>
<td>This reason code is used by RMIService to indicate that the remote method returned an remote object which has not been exported.</td>
</tr>
</tbody>
</table>

Constructors
ServiceException(short reason)
Constructs a ServiceException.
Member Summary

Methods

<table>
<thead>
<tr>
<th>static void</th>
<th>throwIt(short reason)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Throws an instance of ServiceException with the specified reason.</td>
</tr>
</tbody>
</table>

Inherited Member Summary

Methods inherited from interface CardRuntimeException

getReason(), setReason(short)

Methods inherited from class java.lang.Object

close, clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Methods inherited from class java.lang.Throwable

fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString

Fields

ILLEGAL_PARAM

public static final short ILLEGAL_PARAM

This reason code is used to indicate that an input parameter is not allowed.

DISPATCH_TABLE_FULL

public static final short DISPATCH_TABLE_FULL

This reason code is used to indicate that a dispatch table is full

COMMAND_DATA_TOO_LONG

public static final short COMMAND_DATA_TOO_LONG

This reason code is used to indicate that the incoming data for a command in the APDU object does not fit in the APDU buffer.

CANNOT_ACCESS_IN_COMMAND

public static final short CANNOT_ACCESS_IN_COMMAND

This reason code is used to indicate that the command in the APDU object cannot be accessed for input processing.

CANNOT_ACCESS_OUT_COMMAND

public static final short CANNOT_ACCESS_OUT_COMMAND
This reason code is used to indicate that the command in the APDU object cannot be accessed for output processing.

public static final short COMMAND_IS_FINISHED

This reason code is used to indicate that the command in the APDU object has been completely processed.

public static final short REMOTE_OBJECT_NOT_EXPORTED

This reason code is used by RMIService to indicate that the remote method returned an remote object which has not been exported.

Constructors

ServiceException( short )

public ServiceException( short reason )

Constructs a ServiceException.

Parameters:

reason - the reason for the exception.

Methods

throwIt( short )

public static void throwIt( short reason )

Throws an instance of ServiceException with the specified reason.

Parameters:

reason - the reason for the exception.

Throws:

ServiceException - always.
Package javacard.security

Description
Provides Java Card exceptions that may be re-thrown on the client.

## Class Summary

<table>
<thead>
<tr>
<th>Exceptions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CryptoException</td>
<td>CryptoException represents a cryptography-related exception.</td>
</tr>
</tbody>
</table>
javacard.security

CryptoException

Declaration

public class CryptoException extends CardRuntimeException

java.lang.Object
   +--java.lang.Throwable
      +--java.lang.Exception
         +--java.lang.RuntimeException
            +-- javacard.framework.CardRuntimeException
               +--javacard.security.CryptoException

All Implemented Interfaces: java.io.Serializable

Direct Known Subclasses: CryptoExceptionSubclass

Description

CryptoException represents a cryptography-related exception.

Member Summary

<table>
<thead>
<tr>
<th>Fields</th>
</tr>
</thead>
</table>
| static short ILLEGAL_USE
   This reason code is used to indicate that the signature or cipher algorithm does not pad the incoming message and the input message is not block aligned. |
| static short ILLEGAL_VALUE
   This reason code is used to indicate that one or more input parameters is out of allowed bounds. |
| static short INVALID_INIT
   This reason code is used to indicate that the signature or cipher object has not been correctly initialized for the requested operation. |
| static short NO_SUCH_ALGORITHM
   This reason code is used to indicate that the requested algorithm or key type is not supported. |
| static short UNINITIALIZED_KEY
   This reason code is used to indicate that the key is uninitialized. |

<table>
<thead>
<tr>
<th>Constructors</th>
</tr>
</thead>
</table>
| CryptoException(short reason)
   Constructs a CryptoException with the specified reason. |

<table>
<thead>
<tr>
<th>Methods</th>
</tr>
</thead>
</table>
| static void throwIt(short reason)
   Throws an instance of CryptoException with the specified reason. |
Inherited Member Summary

Methods inherited from interface CardRuntimeException
getReason(), setReason(short)

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Methods inherited from class java.lang.Throwable
fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace,
printStackTrace, toString

Fields

ILLEGAL_VALUE

public static final short ILLEGAL_VALUE
This reason code is used to indicate that one or more input parameters is out of allowed bounds.

UNINITIALIZED_KEY

public static final short UNINITIALIZED_KEY
This reason code is used to indicate that the key is uninitialized.

NO_SUCH_ALGORITHM

public static final short NO_SUCH_ALGORITHM
This reason code is used to indicate that the requested algorithm or key type is not supported.

INVALID_INIT

public static final short INVALID_INIT
This reason code is used to indicate that the signature or cipher object has not been correctly initialized for the requested operation.

ILLEGAL_USE

public static final short ILLEGAL_USE
This reason code is used to indicate that the signature or cipher algorithm does not pad the incoming message and the input message is not block aligned.
Constructors

CryptoException(short)

```java
public CryptoException(short reason)

Constructs a CryptoException with the specified reason. To conserve on resources use throwIt() to use the JCRE owned instance of this class.
```

**Parameters:**

- `reason` - the reason for the exception.

Methods

throwIt(short)

```java
public static void throwIt(short reason)

Throws an instance of CryptoException with the specified reason.
```

**Parameters:**

- `reason` - the reason for the exception.

**Throws:**

- `CryptoException` - always.
Index

A
APDUException
   of javacard.framework, 44
APDUException(short)
   of javacard.framework.APDUException, 46
APDUExceptionSubclass
   of com.sun.javacard.javax.smartcard.rmiclient, 14
APDUExceptionSubclass(String, short)
   of com.sun.javacard.javax.smartcard.rmiclient.APDUExceptionSubclass, 15

B
BAD_LENGTH
   of javacard.framework.APDUException, 45
BUFFER_BOUNDS
   of javacard.framework.APDUException, 45
BUFFER_FULL
   of javacard.framework.TransactionException, 59

C
CANNOT_ACCESS_IN_COMMAND
   of javacard.framework.ServiceException, 65
CANNOT_ACCESS_OUT_COMMAND
   of javacard.framework.service.ServiceException, 65
CardAccessor
   of com.sun.javacard.javax.smartcard.rmiclient.CardObjectFactory, 21
CardAccessor Interface, 2
CardException
   of javacard.framework, 47
CardException(short)
   of javacard.framework.CardException, 48
CardExceptionSubclass
   of com.sun.javacard.javax.smartcard.rmiclient.CardExceptionSubclass, 19
CardExceptionSubclass(String, short)
   of com.sun.javacard.javax.smartcard.rmiclient.CardExceptionSubclass, 19
CardObjectFactory
   of com.sun.javacard.javax.smartcard.rmiclient.CardObjectFactory, 21
CardRuntimeException
   of javacard.framework.CardRuntimeException(short)
   of javacard.framework.CardRuntimeException, 50
CardRuntimeExceptionSubclass
   of com.sun.javacard.javax.smartcard.rmiclient.CardRuntimeExceptionSubclass, 25
com.sun.javacard.javax.smartcard.rmiclient package, 13
COMMAND_DATA_TOO_LONG
of javacard.framework.service.
ServiceException, 65
COMMAND_IS_FINISHED
of javacard.framework.service.
ServiceException, 66
CryptoException
of javacard.security, 68
CryptoException(short)
of javacard.security.CryptoException, 70
CryptoExceptionSubclass
of com.sun.javacard.java.smartcard.rmiclient, 26
CryptoExceptionSubclass(String, short)
of com.sun.javacard.java.smartcard.rmiclient.
CryptoExceptionSubclass, 27

D
DISPATCH_TABLE_FULL
of javacard.framework.service.
ServiceException, 65

E
exception handling, 5
exchangeAPDU(byte[])
of com.sun.javacard.java.smartcard.rmiclient.
CardAccessor, 16

G
getInitialReference()
of com.sun.javacard.java.smartcard.rmiclient.
JavaCardRMIConnect, 32
getINSByte()
of com.sun.javacard.java.smartcard.rmiclient.
CardObjectFactory, 22
getMessage()
of com.sun.javacard.java.smartcard.rmiclient.
APDUExceptionSubclass, 15
of com.sun.javacard.java.smartcard.rmiclient.
ExceptionSubclass, 19
of com.sun.javacard.java.smartcard.rmiclient.
CardRuntimeExceptionSubclass, 25
of com.sun.javacard.java.smartcard.rmiclient.
CryptoExceptionSubclass, 27
of com.sun.javacard.java.smartcard.rmiclient.
ISOExceptionSubclass, 29
of com.sun.javacard.java.smartcard.rmiclient.
PINExceptionSubclass, 34
of com.sun.javacard.java.smartcard.rmiclient.
ServiceExceptionSubclass, 36
of com.sun.javacard.java.smartcard.rmiclient.
SystemExceptionSubclass, 38
of com.sun.javacard.java.smartcard.rmiclient.
TransactionExceptionSubclass, 40
of com.sun.javacard.java.smartcard.rmiclient.
UserExceptionSubclass, 42
getObject(byte[], int, Class)
of com.sun.javacard.java.smartcard.rmiclient.
CardObjectFactory, 22
getReason()
of javacard.framework.CardException, 48
of javacard.framework.
CardRuntimeException, 50
getRemoteObject(byte[], int)
of com.sun.javacard.java.smartcard.rmiclient.
CardObjectFactory, 22
getRemoteRefFormat()
of com.sun.javacard.java.smartcard.rmiclient.
CardObjectFactory, 23
getSessionIdentifier()
of com.sun.javacard.java.smartcard.rmiclient.
CardAccessor, 16

I
ILLEGAL_AID
of javacard.framework.SystemException, 56
ILLEGAL_PARAM
of javacard.framework.service.
ServiceException, 65
ILLEGAL_TRANSIENT
of javacard.framework.SystemException, 56
ILLEGAL_USE
of javacard.framework.APDUException, 45
of javacard.framework.SystemException, 57
of javacard.security.CryptoException, 69
ILLEGAL_VALUE
of javacard.framework.PINException, 54
of javacard.framework.SystemException, 56
of javacard.security.CryptoException, 69
IN_PROGRESS
of javacard.framework.
TransactionException, 59
interfaces, for client applications, 2
INTERNAL_FAILURE
of javacard.framework.
    TransactionException, 59
INVALID_INIT
    of javacard.security.CryptoException, 69
IO_ERROR
    of javacard.framework.APDUException, 46
ISOException
    of javacard.framework, 51
ISOException(short)
    of javacard.framework.ISOException, 52
ISOExceptionSubclass
    of com.sun.javacard.javax.smartcard.
        rmiclient, 28
ISOExceptionSubclass(String, short)
    of com.sun.javacard.javax.smartcard.rmiclient.
        PINExceptionSubclass, 29
J
Java Card RMI, features, 1
javacard.framework
    package, 43
javacard.framework.service
    package, 63
javacard.security
    package, 67
JavaCardRMIConnect
    of com.sun.javacard.javax.smartcard.
        rmiclient, 30
JavaCardRMIConnect class, 2
JavaCardRMIConnect(CardAccessor)
    of com.sun.javacard.javax.smartcard.rmiclient.
        JavaCardRMIConnect, 31
P
PINException
    of javacard.framework, 53
PINException(short)
    of javacard.framework.PINException, 54
PINExceptionSubclass
    of com.sun.javacard.javax.smartcard.
        rmiclient, 33
PINExceptionSubclass(String, short)
    of com.sun.javacard.javax.smartcard.rmiclient.
        PINExceptionSubclass, 34
proxy generation, 4
R
REF_FORMAT_CLASS
    of com.sun.javacard.javax.smartcard.rmiclient.
        CardObjectFactory, 21
REF_FORMAT_INTERFACES
    of com.sun.javacard.javax.smartcard.rmiclient.
        CardObjectFactory, 21
REF_FORMAT_NONE
    of com.sun.javacard.javax.smartcard.rmiclient.
        CardObjectFactory, 21
REMOTE_OBJECT_NOT_EXPORTED
    of javacard.framework.service.
        ServiceException, 66
S
selectApplet(byte[])
    of com.sun.javacard.javax.smartcard.rmiclient.
        JavaCardRMIConnect, 31
selectApplet(byte[], CardObjectFactory)
    of com.sun.javacard.javax.smartcard.rmiclient.
        JavaCardRMIConnect, 31
ServiceException
    of javacard.framework.service, 64
ServiceException(short)
    of javacard.framework.service.
        ServiceException, 66
ServiceExceptionSubclass
    of com.sun.javacard.javax.smartcard.
        rmiclient, 35
ServiceExceptionSubclass(String, short)
    of com.sun.javacard.javax.smartcard.rmiclient.
        ServiceExceptionSubclass, 36
setINSByte(byte)

Index  73
stub generation, 4
SystemException
of javacard.framework, 55
SystemException(short)
of javacard.framework.SystemException, 57
SystemExceptionSubclass
of com.sun.javacard.javas.card.rmiclient.
SystemExceptionSubclass, 37
SystemExceptionSubclass(String, short)
of com.sun.javacard.javas.card.rmiclient.
SystemExceptionSubclass, 38

UNINITIALIZED_KEY
of javacard.security.CryptoException, 69
UserException
of javacard.framework, 61
UserException()
of javacard.framework.UserException, 62
UserException(short)
of javacard.framework.UserException, 62
UserExceptionSubclass
of com.sun.javacard.javas.card.rmiclient.
UserExceptionSubclass, 41
UserExceptionSubclass(String, short)
of com.sun.javacard.javas.card.rmiclient.
UserExceptionSubclass, 42

T
T1_IFD_ABORT
of javacard.framework.APDUException, 46
throwIt(short)
of javacard.framework.APDUException, 46
of javacard.framework.CardException, 48
of javacard.framework.
CardRuntimeException, 50
of javacard.framework.ISOException, 52
of javacard.framework.service.
ServiceException, 66
of javacard.framework.TransactionException, 60
of javacard.framework.UserException, 62
of javacard.security.CryptoException, 70
TransactionException
of javacard.framework, 58
TransactionException(short)
of javacard.framework.
TransactionException, 59
TransactionExceptionSubclass
of com.sun.javacard.javas.card.rmiclient.
TransactionExceptionSubclass, 40
transport layer, securing, 3