

Packages
[com.sun.javacard.clientlib](#)
[com.sun.javacard.](#)
[rmiclientlib](#)

All Classes

[ApduIOCardAccessor](#)
[CardAccessor](#)
[JCCardObjectFactory](#)
[JCCardProxyFactory](#)
[JCRemoteRefImpl](#)
[JCRMIConnect](#)

Packages

<u>com.sun.</u> <u>javacard.</u> <u>clientlib</u>	Provides a framework for building client applications capable of exchanging APDUs with Java Cards.
<u>com.sun.</u> <u>javacard.</u> <u>rmiclientlib</u>	Provides a framework of classes and interfaces for building Java Card RMI-based client applications.

All Classes

[ApduIOCardAccessor](#)

[CardAccessor](#)

[JCCardObjectFactory](#)

[JCCardProxyFactory](#)

[JCRemoteRefImpl](#)

[JCRMIConnect](#)

[com.sun.javacard.clientlib](#)

Interfaces

[*CardAccessor*](#)

Classes

[ApduIOCardAccessor](#)

[com.sun.javacard.rmiclientlib](#)

Classes

[JCCardObjectFactory](#)

[JCCardProxyFactory](#)

[JCRemoteRefImpl](#)

[JCRMIConnect](#)

com.sun.javacard.clientlib

Class ApduIOCardAccessor



All Implemented Interfaces:

[CardAccessor](#)

public class ApduIOCardAccessor

extends java.lang.Object

implements [CardAccessor](#)

Implementation of CardAccessor using ApduIO library

Field Summary	
<div>protected com. sun.javacard. apduio. CadClientInterface</div>	<div>cad</div> <div>Reference to underlying ApduIO object.</div>

Constructor Summary
<div>ApduIOCardAccessor ()</div> <div>Creates a new instance of ApduIOCardAccessor</div>

Method Summary

void	<code>closeCard()</code> Close and powerdown the card.
byte []	<code>exchangeAPDU(byte[] capdu)</code> Implementation of exchangeAPDU method of CardAccessor interface

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

Field Detail

cad

protected com.sun.javacard.apduio.CadClientInterface **cad**

Reference to underlying AduIO object.

Constructor Detail

ApduIOCardAccessor

public **ApduIOCardAccessor()**
throws java.lang.Exception

Creates a new instance of ApduIOCardAccessor

Throws:

java.lang.Exception - Thrown is config file jcclient.properties is not found in classpath or communication error happened.

Method Detail

exchangeAPDU

public byte[] **exchangeAPDU**(byte[] capdu)

throws `java.io.IOException`

Implementation of `exchangeAPDU` method of `CardAccessor` interface

Specified by:

[exchangeAPDU](#) in interface [CardAccessor](#)

Parameters:

`capdu` - byte array containing C-APDU

Returns:

R-APDU in the format described in the javadoc for `CardAccessor`

Throws:

`java.io.IOException` - In case of I/O error

closeCard

```
public void closeCard()  
    throws java.lang.Exception
```

Close and powerdown the card.

Specified by:

[closeCard](#) in interface [CardAccessor](#)

Throws:

`java.lang.Exception` - Thrown if a problem occurred

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

com.sun.javacard.clientlib

Interface CardAccessor

All Known Implementing Classes:

[AduIOCardAccessor](#)

```
public interface CardAccessor
```

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.

Method Summary	
void	closeCard () This method closes and resets the card
byte []	exchangeAPDU (byte[] sendData) This method sends the specified data to the smartcard, waits for the response and returns the response in the return data.

Method Detail

exchangeAPDU

```
byte[] exchangeAPDU(byte[] sendData)  
           throws java.io.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:

`java.io.IOException` - if communication error occurs

closeCard

```
void closeCard()  
    throws java.lang.Exception
```

This method closes and resets the card

Throws:

`java.lang.Exception` - Exception with a message about the problem.

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

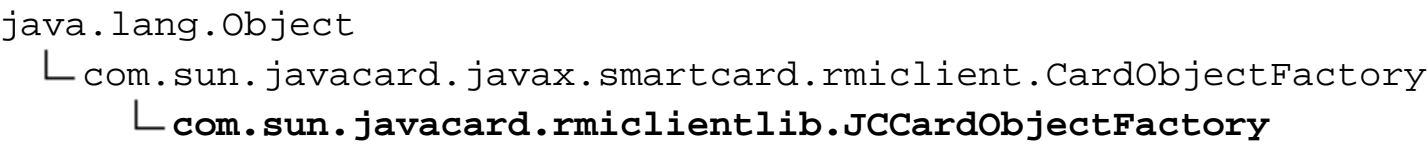
[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

com.sun.javacard.rmiclientlib

Class JCCardObjectFactory



public class **JCCardObjectFactory**

extends com.sun.javacard.javax.smartcard.rmiclient.CardObjectFactory

Processes the data returned from the card in the format defined for Java Card RMI. Object references must contain class names. Extends CardObjectFactory.

Field Summary

Fields inherited from class com.sun.javacard.javax.smartcard.rmiclient.CardObjectFactory
REF_FORMAT_CLASS, REF_FORMAT_INTERFACES, REF_FORMAT_NONE

Constructor Summary

JCCardObjectFactory ()
The constructor.

Method Summary

protected java. rmi. Remote	getRemoteObject (byte[] buffer, int tagOffset, CardAccessor cardAccessor) Creates the stub instance for object reference returned from the card, assuming the card returned a reference with class name.
byte	getRemoteRefFormat () Returns constant REF_FORMAT_CLASS defined in class CardObjectFactory.

Methods inherited from class com.sun.javacard.javax.smartcard.rmiclient.CardObjectFactory

getINSByte, getObject, setINSByte

Methods inherited from class java.lang.Object

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

Constructor Detail

JCCardObjectFactory

```
public JCCardObjectFactory()
```

The constructor.

Method Detail

getRemoteObject

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

Creates the stub instance for object reference returned from the card, assuming the card returned a reference with class name.

Specified by:

`getRemoteObject` in class `com.sun.javacard.javax.smartcard.
rmiclient.CardObjectFactory`

Parameters:

`cardAccessor` - used to instantiate stubs of remote objects
`buffer` - APDU buffer
`tagOffset` - Offset to tag

Returns:

The resulting stub.

Throws:

`java.lang.Exception` - Failed to instantiate a stub

getRemoteRefFormat

```
public byte getRemoteRefFormat()
```

Returns constant `REF_FORMAT_CLASS` defined in class `CardObjectFactory`.

Specified by:

`getRemoteRefFormat` in class `com.sun.javacard.javax.smartcard.
rmiclient.CardObjectFactory`

Returns:

`REF_FORMAT_CLASS` value defined above

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

com.sun.javacard.rmiclientlib

Class JCCardProxyFactory

```
java.lang.Object
└─ com.sun.javacard.javax.smartcard.rmiclient.CardObjectFactory
    └─ com.sun.javacard.rmiclientlib.JCCardProxyFactory
```

```
public class JCCardProxyFactory

extends com.sun.javacard.javax.smartcard.rmiclient.CardObjectFactory
```

Processes the data returned from the card in the format defined for Java Card RMI. Object references must contain lists of interface names. Extends CardObjectFactory.

Field Summary

Fields inherited from class com.sun.javacard.javax.smartcard.rmiclient.CardObjectFactory

REF_FORMAT_CLASS, REF_FORMAT_INTERFACES, REF_FORMAT_NONE

Constructor Summary

[JCCardProxyFactory](#)()
Constructor for the factory.

Method Summary

protected java. rmi. Remote	<code>getRemoteObject</code> (byte[] buffer, int tagOffset, <code>CardAccessor</code> cardAccessor) Creates the stub instance for object reference returned from the card, assuming the card returned a reference with list of interface names.
byte	<code>getRemoteRefFormat</code> () Returns constant <code>REF_FORMAT_INTERFACES</code> defined in class <code>CardObjectFactory</code> .

Methods inherited from class `com.sun.javacard.javax.smartcard.rmiclient.CardObjectFactory`

`getINSByte`, `getObject`, `setINSByte`

Methods inherited from class `java.lang.Object`

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

Constructor Detail

JCCardProxyFactory

```
public JCCardProxyFactory()
```

Constructor for the factory.

Method Detail

getRemoteRefFormat

```
public byte getRemoteRefFormat()
```

Returns constant `REF_FORMAT_INTERFACES` defined in class `CardObjectFactory`.

Specified by:

`getRemoteRefFormat` in class `com.sun.javacard.javax.smartcard.rmiclient.CardObjectFactory`

Returns:

The format constant.

getRemoteObject

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

Creates the stub instance for object reference returned from the card, assuming the card returned a reference with list of interface names.

Specified by:

`getRemoteObject` in class `com.sun.javacard.javax.smartcard.rmiclient.CardObjectFactory`

Parameters:

`cardAccessor` -
`buffer` - APDU buffer.
`tagOffset` - Offset to tag.

Returns:

The instance of the proxy.

Throws:

`java.lang.Exception` - Thrown if the proxy instance cannot be instantiated

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

com.sun.javacard.rmiclientlib

Class JCRemoteRefImpl



All Implemented Interfaces:

java.io.Externalizable, java.lang.reflect.InvocationHandler, java.rmi.server.RemoteRef

```
public class JCRemoteRefImpl
```

extends java.lang.Object
implements java.rmi.server.RemoteRef, java.lang.reflect.InvocationHandler

Represents a reference to a card object. This class is a Java Card RMI implementation of the RemoteRef interface. It is used in conjunction with Java RMIC generated stubs or dynamically generated proxies for Java Card RMI method invocations.

See Also:

Field Summary

Fields inherited from interface java.rmi.server.RemoteRef	
packagePrefix	serialVersionUID

Constructor Summary

JCRemoteRefImpl (short objID, java.lang.String a_string, CardAccessor ca, com.sun.javacard.javax.smartcard.rmiclient.CardObjectFactory cOF)
Creates new JCRemoteRefImpl

Method Summary

void	done (java.rmi.server.RemoteCall remoteCall)
	Deprecated.

java. lang. String	<code>getRefClass</code> (java.io.ObjectOutput objectOutput) Unsupported operation.
java. lang. Object	<code>invoke</code> (java.lang.Object obj, java.lang.reflect.Method method, java.lang.Object[] params) This method is used by dynamically generated proxies.
void	<code>invoke</code> (java.rmi.server.RemoteCall remoteCall) Deprecated.
java. lang. Object	<code>invoke</code> (java.rmi.Remote remote, java.lang.reflect.Method method, java.lang.Object[] params, long unused) This method is used by rmic-generated stubs.
java.rmi. server. RemoteCall	<code>newCall</code> (java.rmi.server.RemoteObject remoteObject, java.rmi.server.Operation[] operation, int param, long param3) Deprecated.
void	<code>readExternal</code> (java.io.ObjectInput objectInput) Unsupported operation.
boolean	<code>remoteEquals</code> (java.rmi.server.RemoteRef remoteRef) Compares two remote objects for being identical.
int	<code>remoteHashCode</code> () Unsupported operation.
java. lang. String	<code>remoteToString</code> () String representation of remote object.
void	<code>writeExternal</code> (java.io.ObjectOutput objectOutput) Unsupported operation.

Methods inherited from class java.lang.Object

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

Constructor Detail

JCRemoteRefImpl

```
public JCRemoteRefImpl(short objID,
                       java.lang.String a_string,
                       CardAccessor ca,
                       com.sun.javacard.javax.smartcard.rmiclient.
CardObjectFactory cOF)
```

Creates new JCRemoteRefImpl

Parameters:

objID - 2 byte Object ID from card remote reference descriptor
a_string - Anticollision string for the class of the remote object
cA - CardAccessor
cOF - CardObjectFactory

Method Detail

getRefClass

```
public java.lang.String getRefClass(java.io.ObjectOutput objectOutput)
```

Unsupported operation.

Specified by:

getRefClass in interface `java.rmi.server.RemoteRef`

Parameters:

objectOutput -

invoke

```
public java.lang.Object invoke(java.rmi.Remote remote,  
                                java.lang.reflect.Method method,  
                                java.lang.Object[] params,  
                                long unused)  
    throws java.io.IOException,  
           java.rmi.RemoteException,  
           java.lang.Exception
```

This method is used by rmic-generated stubs.

Specified by:

invoke in interface `java.rmi.server.RemoteRef`

Parameters:

remote - Reference to the stub - not used.

method - `java.lang.reflect.Method` object containing information about the method to be invoked.

params - Array of parameters. Primitives are wrapped.

unused - rmic-generated hash of the method. Not used.

Returns:

The result returned from the card.

Throws:

`java.io.IOException` - If a communication error occurred.

`java.rmi.RemoteException` - If an RMI error occurred.

`java.lang.Exception` - Exception corresponding to the one that was thrown on the card.

remoteHashCode

```
public int remoteHashCode()
```

Unsupported operation.

Specified by:

remoteHashCode in interface java.rmi.server.RemoteRef

Returns:

A number which is the same for all objects.

remoteToString

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

String representation of remote object.

Specified by:

remoteToString in interface java.rmi.server.RemoteRef

Returns:

A String representation of the remote object.

readExternal

```
public void readExternal(java.io.ObjectInput objectInput)  
    throws java.io.IOException,  
           java.lang.ClassNotFoundException
```

Unsupported operation.

Specified by:

readExternal in interface java.io.Externalizable

Parameters:

objectInput -

Throws:

java.io.IOException

java.lang.ClassNotFoundException

writeExternal

```
public void writeExternal(java.io.ObjectOutput objectOutput)  
    throws java.io.IOException
```

Unsupported operation.

Specified by:

writeExternal in interface `java.io.Externalizable`

Parameters:

objectOutput -

Throws:

`java.io.IOException`

newCall

```
public java.rmi.server.RemoteCall newCall(java.rmi.server.RemoteObject remoteObject,  
                                             java.rmi.server.Operation[] operation,  
                                             int param,  
                                             long param3)  
    throws java.rmi.RemoteException
```

Deprecated.

Deprecated and not implemented

Specified by:

newCall in interface `java.rmi.server.RemoteRef`

Parameters:

remoteObject -

operation -

param -

param3 -

Throws:

`java.rmi.RemoteException`

invoke

```
public void invoke(java.rmi.server.RemoteCall remoteCall)  
    throws java.lang.Exception
```

Deprecated.

Deprecated and not implemented

Specified by:

invoke in interface `java.rmi.server.RemoteRef`

Parameters:

remoteCall -

Throws:

`java.lang.Exception`

remoteEquals

```
public boolean remoteEquals(java.rmi.server.RemoteRef remoteRef)
```

Compares two remote objects for being identical.

Specified by:

remoteEquals in interface java.rmi.server.RemoteRef

Parameters:

remoteRef - RemoteRef to the other remote object.

Returns:

true if corresponding remote objects are identical.

done

```
public void done(java.rmi.server.RemoteCall remoteCall)  
    throws java.rmi.RemoteException
```

Deprecated.

Deprecated and not implemented

Specified by:

done in interface java.rmi.server.RemoteRef

Parameters:

remoteCall -

Throws:

java.rmi.RemoteException

invoke

```
public java.lang.Object invoke(java.lang.Object obj,  
                                java.lang.reflect.Method method,  
                                java.lang.Object[] params)  
    throws java.io.IOException,  
           java.rmi.RemoteException,  
           java.lang.Throwable
```

This method is used by dynamically generated proxies.

Specified by:

invoke in interface java.lang.reflect.InvocationHandler

Parameters:

obj - The reference to the Proxy - not used.

method - Method object containing information about the method.

params - Array of parameters for the method.

Returns:

The result returned from the card.

Throws:

`java.io.IOException` - If a communication error occurred.

`java.rmi.RemoteException` - If an RMI error occurred.

`java.lang.Throwable` - Exception corresponding to the one that was thrown on the card.

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

com.sun.javacard.rmiclientlib

Class JCRMIConnect



```
public class JCRMIConnect
```

extends java.lang.Object

The main class of the Java Card RMI client API. Provides functionality to select a card applet and to obtain an initial reference.

Field Summary	
protected byte	format Format of the remote references.
static byte	REF_WITH_CLASS_NAME Constant used as the 2nd parameter to selectApplet method.
static byte	REF_WITH_INTERFACE_NAMES Constant used as the 2nd parameter to selectApplet method.
protected byte[]	selectResponse Response to the SELECT command is stored in this field.

Constructor Summary	
JCRMIConnect	(CardAccessor ca)
Creates a new instance of JCRMIConnect	

Method Summary	
java. rmi. Remote	getInitialReference () Parses the R-APDU which was returned during selecting an applet, returns a reference to an initial remote object
byte []	selectApplet (byte[] aid, byte format) Selects an applet, requesting initial reference in the format specified by the 2nd parameter.

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

Field Detail

selectResponse

protected byte[] **selectResponse**

Response to the SELECT command is stored in this field.

format

protected byte **format**

Format of the remote references.

REF_WITH_CLASS_NAME

public static final byte **REF_WITH_CLASS_NAME**

Constant used as the 2nd parameter to selectApplet method.

See Also:

[Constant Field Values](#)

REF_WITH_INTERFACE_NAMES

```
public static final byte REF_WITH_INTERFACE_NAMES
```

Constant used as the 2nd parameter to selectApplet method.

See Also:

[Constant Field Values](#)

Constructor Detail

JCRMIConnect

```
public JCRMIConnect(CardAccessor ca)
```

Creates a new instance of JCRMIConnect

Parameters:

ca - Implementaion of a CardAccessor

Method Detail

getInitialReference

```
public java.rmi.Remote getInitialReference()  
                                throws java.lang.Exception
```

Parses the R-APDU which was returned during selecting an applet, returns a reference to an initial remote object

Returns:

Stub or proxy for the initial remote object

Throws:

`java.lang.Exception` - If a problem occurred

selectApplet

```
public byte[] selectApplet(byte[] aid,  
                             byte format)  
    throws java.lang.Exception
```

Selects an applet, requesting initial reference in the format specified by the 2nd parameter. The R-APDU is returned and also stored internally for further processing by the `getInitialReference()` method.

Parameters:

`aid` - AID of the applet to be selected

`format` - Format of the remote references

Returns:

R-APDU

Throws:

`java.lang.Exception` - If a problem occurred

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

Hierarchy For All Packages

Package Hierarchies:

[com.sun.javacard.clientlib](#), [com.sun.javacard.rmiclientlib](#)

Class Hierarchy

- java.lang.Object
 - com.sun.javacard.clientlib.[AduIOCardAccessor](#) (implements com.sun.javacard.clientlib.[CardAccessor](#))
 - com.sun.javacard.javax.smartcard.rmiclient.CardObjectFactory
 - com.sun.javacard.rmiclientlib.[JCCardObjectFactory](#)
 - com.sun.javacard.rmiclientlib.[JCCardProxyFactory](#)
 - com.sun.javacard.rmiclientlib.[JCRemoteRefImpl](#) (implements java.lang.reflect.InvocationHandler, java.rmi.server.RemoteRef)
 - com.sun.javacard.rmiclientlib.[JCRMIConnect](#)

Interface Hierarchy

- com.sun.javacard.clientlib.[CardAccessor](#)

Deprecated API

Contents

- [Deprecated Methods](#)

Deprecated Methods

[com.sun.javacard.rmiclientlib.JCRemoteRefImpl.done\(RemoteCall\)](#)

[com.sun.javacard.rmiclientlib.JCRemoteRefImpl.invoke\(RemoteCall\)](#)

[com.sun.javacard.rmiclientlib.JCRemoteRefImpl.newCall\(RemoteObject, Operation\[\], int, long\)](#)

A

[ApduIOCardAccessor](#) - Class in [com.sun.javacard.clientlib](#)

Implementation of CardAccessor using ApduIO library

[ApduIOCardAccessor\(\)](#) - Constructor for class com.sun.javacard.clientlib.[ApduIOCardAccessor](#)

Creates a new instance of ApduIOCardAccessor

C

[cad](#) - Variable in class com.sun.javacard.clientlib.[ApduIOCardAccessor](#)

Reference to underlying ApduIO object.

[CardAccessor](#) - Interface in [com.sun.javacard.clientlib](#)

The CardAccessor interface represents a generic smartcard communication API.

[closeCard\(\)](#) - Method in class com.sun.javacard.clientlib.[ApduIOCardAccessor](#)

Close and powerdown the card.

[closeCard\(\)](#) - Method in interface com.sun.javacard.clientlib.[CardAccessor](#)

This method closes and resets the card

[com.sun.javacard.clientlib](#) - package com.sun.javacard.clientlib

Provides a framework for building client applications capable of exchanging APDUs with Java Cards.

[com.sun.javacard.rmiclientlib](#) - package com.sun.javacard.rmiclientlib

Provides a framework of classes and interfaces for building Java Card RMI-based client applications.

D

[done\(RemoteCall\)](#) - Method in class com.sun.javacard.rmiclientlib.[JCRemoteRefImpl](#)

Deprecated.

E

[exchangeAPDU\(byte\[\]\)](#) - Method in class com.sun.javacard.clientlib.[ApduIOCardAccessor](#)

Implementation of exchangeAPDU method of CardAccessor interface

[exchangeAPDU\(byte\[\]\)](#) - Method in interface com.sun.javacard.clientlib.[CardAccessor](#)

This method sends the specified data to the smartcard, waits for the response and returns the response in the return data.

F

[format](#) - Variable in class com.sun.javacard.rmiclientlib.[JCRMIConnect](#)

Format of the remote references.

G

[getInitialReference\(\)](#) - Method in class com.sun.javacard.rmiclientlib.[JCRMIConnect](#)

Parses the R-APDU which was returned during selecting an applet, returns a reference to an initial remote object

[getRefClass\(ObjectOutput\)](#) - Method in class com.sun.javacard.rmiclientlib.[JCRemoteRefImpl](#)

Unsupported operation.

[getRemoteObject\(byte\[\], int, CardAccessor\)](#) - Method in class com.sun.javacard.rmiclientlib.[JCCardObjectFactory](#)

Creates the stub instance for object reference returned from the card, assuming the card returned a reference with class name.

[getRemoteObject\(byte\[\], int, CardAccessor\)](#) - Method in class com.sun.javacard.rmiclientlib.[JCCardProxyFactory](#)

Creates the stub instance for object reference returned from the card, assuming the card returned a reference with list of interface names.

[getRemoteRefFormat\(\)](#) - Method in class com.sun.javacard.rmiclientlib.[JCCardObjectFactory](#)

Returns constant REF_FORMAT_CLASS defined in class CardObjectFactory.

[getRemoteRefFormat\(\)](#) - Method in class com.sun.javacard.rmiclientlib.[JCCardProxyFactory](#)

Returns constant REF_FORMAT_INTERFACES defined in class CardObjectFactory.

I

[**invoke\(Remote, Method, Object\[\], long\)**](#) - Method in class [com.sun.javacard.rmiclientlib.JCRemoteRefImpl](#)

This method is used by rmic-generated stubs.

[**invoke\(RemoteCall\)**](#) - Method in class [com.sun.javacard.rmiclientlib.JCRemoteRefImpl](#)

Deprecated.

[**invoke\(Object, Method, Object\[\]\)**](#) - Method in class [com.sun.javacard.rmiclientlib.JCRemoteRefImpl](#)

This method is used by dynamically generated proxies.

J

[**JCCardObjectFactory**](#) - Class in [com.sun.javacard.rmiclientlib](#)

Processes the data returned from the card in the format defined for Java Card RMI.

[**JCCardObjectFactory\(\)**](#) - Constructor for class [com.sun.javacard.rmiclientlib.JCCardObjectFactory](#)

The constructor.

[**JCCardProxyFactory**](#) - Class in [com.sun.javacard.rmiclientlib](#)

Processes the data returned from the card in the format defined for Java Card RMI.

[**JCCardProxyFactory\(\)**](#) - Constructor for class [com.sun.javacard.rmiclientlib.JCCardProxyFactory](#)

Constructor for the factory.

[**JCRemoteRefImpl**](#) - Class in [com.sun.javacard.rmiclientlib](#)

Represents a reference to a card object.

[**JCRemoteRefImpl\(short, String, CardAccessor, CardObjectFactory\)**](#) - Constructor for class [com.sun.javacard.rmiclientlib.JCRemoteRefImpl](#)

Creates new JCRemoteRefImpl

[**JCRMICConnect**](#) - Class in [com.sun.javacard.rmiclientlib](#)

The main class of the Java Card RMI client API.

[**JCRMICConnect\(CardAccessor\)**](#) - Constructor for class [com.sun.javacard.rmiclientlib.JCRMICConnect](#)

Creates a new instance of JCRMICConnect

N

[**newCall\(RemoteObject, Operation\[\], int, long\)**](#) - Method in class [com.sun.javacard.rmiclientlib.JCRemoteRefImpl](#)

Deprecated.

R

[**readExternal\(ObjectInput\)**](#) - Method in class com.sun.javacard.rmiclientlib.[JCRemoteRefImpl](#)

Unsupported operation.

[**REF_WITH_CLASS_NAME**](#) - Static variable in class com.sun.javacard.rmiclientlib.[JCRMIConnect](#)

Constant used as the 2nd parameter to selectApplet method.

[**REF_WITH_INTERFACE_NAMES**](#) - Static variable in class com.sun.javacard.rmiclientlib.

[JCRMIConnect](#)

Constant used as the 2nd parameter to selectApplet method.

[**remoteEquals\(RemoteRef\)**](#) - Method in class com.sun.javacard.rmiclientlib.[JCRemoteRefImpl](#)

Compares two remote objects for being identical.

[**remoteHashCode\(\)**](#) - Method in class com.sun.javacard.rmiclientlib.[JCRemoteRefImpl](#)

Unsupported operation.

[**remoteToString\(\)**](#) - Method in class com.sun.javacard.rmiclientlib.[JCRemoteRefImpl](#)

String representation of remote object.

S

[**selectApplet\(byte\[\], byte\)**](#) - Method in class com.sun.javacard.rmiclientlib.[JCRMIConnect](#)

Selects an applet, requesting initial reference in the format specified by the 2nd parameter.

[**selectResponse**](#) - Variable in class com.sun.javacard.rmiclientlib.[JCRMIConnect](#)

Response to the SELECT command is stored in this field.

W

[**writeExternal\(ObjectOutput\)**](#) - Method in class com.sun.javacard.rmiclientlib.[JCRemoteRefImpl](#)

Unsupported operation.

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [J](#) [N](#) [R](#) [S](#) [W](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV](#) [NEXT](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

How This API Document Is Organized

This API (Application Programming Interface) document has pages corresponding to the items in the navigation bar, described as follows.

Overview

The [Overview](#) page is the front page of this API document and provides a list of all packages with a summary for each. This page can also contain an overall description of the set of packages.

Package

Each package has a page that contains a list of its classes and interfaces, with a summary for each. This page can contain four categories:

- Interfaces (*italic*)
- Classes
- Enums
- Exceptions
- Errors
- Annotation Types

Class/Interface

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three sections consisting of a class/interface description, summary tables, and detailed member descriptions:

- Class inheritance diagram
- Direct Subclasses
- All Known Subinterfaces
- All Known Implementing Classes
- Class/interface declaration
- Class/interface description

- Nested Class Summary
 - Field Summary
 - Constructor Summary
 - Method Summary
-
- Field Detail
 - Constructor Detail
 - Method Detail

Each summary entry contains the first sentence from the detailed description for that item. The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

Annotation Type

Each annotation type has its own separate page with the following sections:

- Annotation Type declaration
- Annotation Type description
- Required Element Summary
- Optional Element Summary
- Element Detail

Enum

Each enum has its own separate page with the following sections:

- Enum declaration
- Enum description
- Enum Constant Summary
- Enum Constant Detail

Tree (Class Hierarchy)

There is a [Class Hierarchy](#) page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. The classes are organized by inheritance structure starting with `java.lang.Object`. The interfaces do not inherit from `java.lang.Object`.

- When viewing the Overview page, clicking on "Tree" displays the hierarchy for all

packages.

- When viewing a particular package, class or interface page, clicking "Tree" displays the hierarchy for only that package.

Deprecated API

The [Deprecated API](#) page lists all of the API that have been deprecated. A deprecated API is not recommended for use, generally due to improvements, and a replacement API is usually given. Deprecated APIs may be removed in future implementations.

Index

The [Index](#) contains an alphabetic list of all classes, interfaces, constructors, methods, and fields.

Prev/Next

These links take you to the next or previous class, interface, package, or related page.

Frames/No Frames

These links show and hide the HTML frames. All pages are available with or without frames.

Constant Field Values

The [Constant Field Values](#) page lists the static final fields and their values.

This help file applies to API documentation generated using the standard doclet.

Overview	Package	Class	Tree	Deprecated	Index	Help
--------------------------	-------------------------	-----------------------	----------------------	----------------------------	-----------------------	----------------------

PREV	NEXT	FRAMES	NO FRAMES	All Classes
----------------------	----------------------	------------------------	---------------------------	-----------------------------

Packages

com.sun.javacard.clientlib	Provides a framework for building client applications capable of exchanging APDUs with Java Cards.
com.sun.javacard.rmiclientlib	Provides a framework of classes and interfaces for building Java Card RMI-based client applications.

Package com.sun.javacard.clientlib

Provides a framework for building client applications capable of exchanging APDUs with Java Cards.

See: [Description](#)

Interface Summary	
CardAccessor	The CardAccessor interface represents a generic smartcard communication API.

Class Summary	
AduIOCardAccessor	Implementation of CardAccessor using AduIO library

Package com.sun.javacard.clientlib Description

Provides a framework for building client applications capable of exchanging APDUs with Java Cards.

Package com.sun.javacard.rmiclientlib

Provides a framework of classes and interfaces for building Java Card RMI-based client applications.

See: [Description](#)

Class Summary	
JCCardObjectFactory	Processes the data returned from the card in the format defined for Java Card RMI.
JCCardProxyFactory	Processes the data returned from the card in the format defined for Java Card RMI.
JCRemoteRefImpl	Represents a reference to a card object.
JCRMIConnect	The main class of the Java Card RMI client API.

Package com.sun.javacard.rmiclientlib Description

Provides a framework of classes and interfaces for building Java Card RMI-based client applications.

Hierarchy For Package com.sun.javacard.clientlib

Package Hierarchies:

[All Packages](#)

Class Hierarchy

- java.lang.Object
 - com.sun.javacard.clientlib.[AduIOCardAccessor](#) (implements com.sun.javacard.clientlib.[CardAccessor](#))

Interface Hierarchy

- com.sun.javacard.clientlib.[CardAccessor](#)

All Classes

[ApduIOCardAccessor](#)

[CardAccessor](#)

[JCCardObjectFactory](#)

[JCCardProxyFactory](#)

[JCRemoteRefImpl](#)

[JCRMIConnect](#)

Hierarchy For Package com.sun.javacard.rmiclientlib

Package Hierarchies:

[All Packages](#)

Class Hierarchy

- java.lang.Object
 - com.sun.javacard.javax.smartcard.rmiclient.CardObjectFactory
 - com.sun.javacard.rmiclientlib.[JCCardObjectFactory](#)
 - com.sun.javacard.rmiclientlib.[JCCardProxyFactory](#)
 - com.sun.javacard.rmiclientlib.[JCRemoteRefImpl](#) (implements java.lang.reflect.InvocationHandler, java.rmi.server.RemoteRef)
 - com.sun.javacard.rmiclientlib.[JCRMIConnect](#)

Constant Field Values

Contents

- [com.sun.*](#)

com.sun.*

com.sun.javacard.rmiclientlib.[JCRMIConnect](#)

public static final byte	REF WITH CLASS NAME	0
public static final byte	REF WITH INTERFACE NAMES	16

Copyright © 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements. Use is subject to license terms.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, Sparc, Java Card, Java Developer Connection, Javadoc, JDK, JVM, J2ME, NetBeans and J2SE are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Products covered by and information contained in this service manual are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright © 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuels relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains listés à l'adresse <http://www.sun.com/patents> et un ou les brevets supplémentaires ou les applications de brevet en attente aux Etats - Unis et dans les autres pays. L'utilisation est soumise aux termes de la Licence.

Cette distribution peut comprendre des composants développés par des tierces parties.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, Solaris, Sparc, Java Card, Java Developer Connection, Javadoc, JDK, JVM, J2ME, NetBeans et J2SE sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Les produits qui font l'objet de ce manuel d'entretien et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes biologiques et chimiques ou du nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.