| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/KeyInfoFactory.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfo.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyName.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html)    [**NO FRAMES**](http://docs.google.com/KeyInfoFactory.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **javax.xml.crypto.dsig.keyinfo**

Class KeyInfoFactory

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.xml.crypto.dsig.keyinfo.KeyInfoFactory**

public abstract class **KeyInfoFactory**extends [Object](http://docs.google.com/java/lang/Object.html)

A factory for creating [KeyInfo](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfo.html) objects from scratch or for unmarshalling a KeyInfo object from a corresponding XML representation.

Each instance of KeyInfoFactory supports a specific XML mechanism type. To create a KeyInfoFactory, call one of the static [getInstance](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#getInstance(java.lang.String)) methods, passing in the XML mechanism type desired, for example:

KeyInfoFactory factory = KeyInfoFactory.getInstance("DOM");

The objects that this factory produces will be based on DOM and abide by the DOM interoperability requirements as defined in the  [DOM Mechanism Requirements](http://docs.google.com/technotes/guides/security/xmldsig/overview.html#DOM%20Mechanism%20Requirements) section of the API overview. See the  [Service Providers](http://docs.google.com/technotes/guides/security/xmldsig/overview.html#Service%20Provider) section of the API overview for a list of standard mechanism types.

KeyInfoFactory implementations are registered and loaded using the [Provider](http://docs.google.com/java/security/Provider.html) mechanism. For example, a service provider that supports the DOM mechanism would be specified in the Provider subclass as:

put("KeyInfoFactory.DOM", "org.example.DOMKeyInfoFactory");

Also, the XMLStructures that are created by this factory may contain state specific to the KeyInfo and are not intended to be reusable.

An implementation MUST minimally support the default mechanism type: DOM.

Note that a caller must use the same KeyInfoFactory instance to create the XMLStructures of a particular KeyInfo object. The behavior is undefined if XMLStructures from different providers or different mechanism types are used together.

**Concurrent Access**

The static methods of this class are guaranteed to be thread-safe. Multiple threads may concurrently invoke the static methods defined in this class with no ill effects.

However, this is not true for the non-static methods defined by this class. Unless otherwise documented by a specific provider, threads that need to access a single KeyInfoFactory instance concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating a different KeyInfoFactory instance need not synchronize.

**Since:** 1.6

| **Constructor Summary** | |
| --- | --- |
| protected | [**KeyInfoFactory**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#KeyInfoFactory())()            Default constructor, for invocation by subclasses. |

| **Method Summary** | |
| --- | --- |
| static [KeyInfoFactory](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html) | [**getInstance**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#getInstance())()            Returns a KeyInfoFactory that supports the default XML processing mechanism and representation type ("DOM"). |
| static [KeyInfoFactory](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html) | [**getInstance**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#getInstance(java.lang.String))([String](http://docs.google.com/java/lang/String.html) mechanismType)            Returns a KeyInfoFactory that supports the specified XML processing mechanism and representation type (ex: "DOM"). |
| static [KeyInfoFactory](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html) | [**getInstance**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#getInstance(java.lang.String,%20java.security.Provider))([String](http://docs.google.com/java/lang/String.html) mechanismType, [Provider](http://docs.google.com/java/security/Provider.html) provider)            Returns a KeyInfoFactory that supports the requested XML processing mechanism and representation type (ex: "DOM"), as supplied by the specified provider. |
| static [KeyInfoFactory](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html) | [**getInstance**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#getInstance(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) mechanismType, [String](http://docs.google.com/java/lang/String.html) provider)            Returns a KeyInfoFactory that supports the requested XML processing mechanism and representation type (ex: "DOM"), as supplied by the specified provider. |
| [String](http://docs.google.com/java/lang/String.html) | [**getMechanismType**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#getMechanismType())()            Returns the type of the XML processing mechanism and representation supported by this KeyInfoFactory (ex: "DOM") |
| [Provider](http://docs.google.com/java/security/Provider.html) | [**getProvider**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#getProvider())()            Returns the provider of this KeyInfoFactory. |
| abstract  [URIDereferencer](http://docs.google.com/javax/xml/crypto/URIDereferencer.html) | [**getURIDereferencer**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#getURIDereferencer())()            Returns a reference to the URIDereferencer that is used by default to dereference URIs in [RetrievalMethod](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/RetrievalMethod.html) objects. |
| abstract  boolean | [**isFeatureSupported**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#isFeatureSupported(java.lang.String))([String](http://docs.google.com/java/lang/String.html) feature)            Indicates whether a specified feature is supported. |
| abstract  [KeyInfo](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfo.html) | [**newKeyInfo**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#newKeyInfo(java.util.List))([List](http://docs.google.com/java/util/List.html) content)            Creates a KeyInfo containing the specified list of key information types. |
| abstract  [KeyInfo](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfo.html) | [**newKeyInfo**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#newKeyInfo(java.util.List,%20java.lang.String))([List](http://docs.google.com/java/util/List.html) content, [String](http://docs.google.com/java/lang/String.html) id)            Creates a KeyInfo containing the specified list of key information types and optional id. |
| abstract  [KeyName](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyName.html) | [**newKeyName**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#newKeyName(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Creates a KeyName from the specified name. |
| abstract  [KeyValue](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyValue.html) | [**newKeyValue**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#newKeyValue(java.security.PublicKey))([PublicKey](http://docs.google.com/java/security/PublicKey.html) key)            Creates a KeyValue from the specified public key. |
| abstract  [PGPData](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/PGPData.html) | [**newPGPData**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#newPGPData(byte%5B%5D))(byte[] keyId)            Creates a PGPData from the specified PGP public key identifier. |
| abstract  [PGPData](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/PGPData.html) | [**newPGPData**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#newPGPData(byte%5B%5D,%20byte%5B%5D,%20java.util.List))(byte[] keyId, byte[] keyPacket, [List](http://docs.google.com/java/util/List.html) other)            Creates a PGPData from the specified PGP public key identifier, and optional key material packet and list of external elements. |
| abstract  [PGPData](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/PGPData.html) | [**newPGPData**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#newPGPData(byte%5B%5D,%20java.util.List))(byte[] keyPacket, [List](http://docs.google.com/java/util/List.html) other)            Creates a PGPData from the specified PGP key material packet and optional list of external elements. |
| abstract  [RetrievalMethod](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/RetrievalMethod.html) | [**newRetrievalMethod**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#newRetrievalMethod(java.lang.String))([String](http://docs.google.com/java/lang/String.html) uri)            Creates a RetrievalMethod from the specified URI. |
| abstract  [RetrievalMethod](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/RetrievalMethod.html) | [**newRetrievalMethod**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#newRetrievalMethod(java.lang.String,%20java.lang.String,%20java.util.List))([String](http://docs.google.com/java/lang/String.html) uri, [String](http://docs.google.com/java/lang/String.html) type, [List](http://docs.google.com/java/util/List.html) transforms)            Creates a RetrievalMethod from the specified parameters. |
| abstract  [X509Data](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/X509Data.html) | [**newX509Data**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#newX509Data(java.util.List))([List](http://docs.google.com/java/util/List.html) content)            Creates a X509Data containing the specified list of X.509 content. |
| abstract  [X509IssuerSerial](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/X509IssuerSerial.html) | [**newX509IssuerSerial**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#newX509IssuerSerial(java.lang.String,%20java.math.BigInteger))([String](http://docs.google.com/java/lang/String.html) issuerName, [BigInteger](http://docs.google.com/java/math/BigInteger.html) serialNumber)            Creates an X509IssuerSerial from the specified X.500 issuer distinguished name and serial number. |
| abstract  [KeyInfo](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfo.html) | [**unmarshalKeyInfo**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html#unmarshalKeyInfo(javax.xml.crypto.XMLStructure))([XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html) xmlStructure)            Unmarshals a new KeyInfo instance from a mechanism-specific XMLStructure (ex: [DOMStructure](http://docs.google.com/javax/xml/crypto/dom/DOMStructure.html)) instance. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### KeyInfoFactory

protected **KeyInfoFactory**()

Default constructor, for invocation by subclasses.

| **Method Detail** |
| --- |

### getInstance

public static [KeyInfoFactory](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) mechanismType)

Returns a KeyInfoFactory that supports the specified XML processing mechanism and representation type (ex: "DOM").

This method uses the standard JCA provider lookup mechanism to locate and instantiate a KeyInfoFactory implementation of the desired mechanism type. It traverses the list of registered security Providers, starting with the most preferred Provider. A new KeyInfoFactory object from the first Provider that supports the specified mechanism is returned.

Note that the list of registered providers may be retrieved via the [Security.getProviders()](http://docs.google.com/java/security/Security.html#getProviders()) method.

**Parameters:**mechanismType - the type of the XML processing mechanism and representation. See the  [Service Providers](http://docs.google.com/technotes/guides/security/xmldsig/overview.html#Service%20Provider) section of the API overview for a list of standard mechanism types. **Returns:**a new KeyInfoFactory **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if mechanismType is null [NoSuchMechanismException](http://docs.google.com/javax/xml/crypto/NoSuchMechanismException.html) - if no Provider supports a KeyInfoFactory implementation for the specified mechanism**See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getInstance

public static [KeyInfoFactory](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) mechanismType,  
 [Provider](http://docs.google.com/java/security/Provider.html) provider)

Returns a KeyInfoFactory that supports the requested XML processing mechanism and representation type (ex: "DOM"), as supplied by the specified provider. Note that the specified Provider object does not have to be registered in the provider list.

**Parameters:**mechanismType - the type of the XML processing mechanism and representation. See the  [Service Providers](http://docs.google.com/technotes/guides/security/xmldsig/overview.html#Service%20Provider) section of the API overview for a list of standard mechanism types.provider - the Provider object **Returns:**a new KeyInfoFactory **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if mechanismType or provider are null [NoSuchMechanismException](http://docs.google.com/javax/xml/crypto/NoSuchMechanismException.html) - if a KeyInfoFactory implementation for the specified mechanism is not available from the specified Provider object**See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getInstance

public static [KeyInfoFactory](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) mechanismType,  
 [String](http://docs.google.com/java/lang/String.html) provider)  
 throws [NoSuchProviderException](http://docs.google.com/java/security/NoSuchProviderException.html)

Returns a KeyInfoFactory that supports the requested XML processing mechanism and representation type (ex: "DOM"), as supplied by the specified provider. The specified provider must be registered in the security provider list.

Note that the list of registered providers may be retrieved via the [Security.getProviders()](http://docs.google.com/java/security/Security.html#getProviders()) method.

**Parameters:**mechanismType - the type of the XML processing mechanism and representation. See the  [Service Providers](http://docs.google.com/technotes/guides/security/xmldsig/overview.html#Service%20Provider) section of the API overview for a list of standard mechanism types.provider - the string name of the provider **Returns:**a new KeyInfoFactory **Throws:** [NoSuchProviderException](http://docs.google.com/java/security/NoSuchProviderException.html) - if the specified provider is not registered in the security provider list [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if mechanismType or provider are null [NoSuchMechanismException](http://docs.google.com/javax/xml/crypto/NoSuchMechanismException.html) - if a KeyInfoFactory implementation for the specified mechanism is not available from the specified provider**See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getInstance

public static [KeyInfoFactory](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html) **getInstance**()

Returns a KeyInfoFactory that supports the default XML processing mechanism and representation type ("DOM").

This method uses the standard JCA provider lookup mechanism to locate and instantiate a KeyInfoFactory implementation of the default mechanism type. It traverses the list of registered security Providers, starting with the most preferred Provider. A new KeyInfoFactory object from the first Provider that supports the DOM mechanism is returned.

Note that the list of registered providers may be retrieved via the [Security.getProviders()](http://docs.google.com/java/security/Security.html#getProviders()) method.

**Returns:**a new KeyInfoFactory **Throws:** [NoSuchMechanismException](http://docs.google.com/javax/xml/crypto/NoSuchMechanismException.html) - if no Provider supports a KeyInfoFactory implementation for the DOM mechanism**See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getMechanismType

public final [String](http://docs.google.com/java/lang/String.html) **getMechanismType**()

Returns the type of the XML processing mechanism and representation supported by this KeyInfoFactory (ex: "DOM")

**Returns:**the XML processing mechanism type supported by this KeyInfoFactory

### getProvider

public final [Provider](http://docs.google.com/java/security/Provider.html) **getProvider**()

Returns the provider of this KeyInfoFactory.

**Returns:**the provider of this KeyInfoFactory

### newKeyInfo

public abstract [KeyInfo](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfo.html) **newKeyInfo**([List](http://docs.google.com/java/util/List.html) content)

Creates a KeyInfo containing the specified list of key information types.

**Parameters:**content - a list of one or more [XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html)s representing key information types. The list is defensively copied to protect against subsequent modification. **Returns:**a KeyInfo **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if content is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if content is empty [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if content contains any entries that are not of type [XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html)

### newKeyInfo

public abstract [KeyInfo](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfo.html) **newKeyInfo**([List](http://docs.google.com/java/util/List.html) content,  
 [String](http://docs.google.com/java/lang/String.html) id)

Creates a KeyInfo containing the specified list of key information types and optional id. The id parameter represents the value of an XML ID attribute and is useful for referencing the KeyInfo from other XML structures.

**Parameters:**content - a list of one or more [XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html)s representing key information types. The list is defensively copied to protect against subsequent modification.id - the value of an XML ID (may be null) **Returns:**a KeyInfo **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if content is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if content is empty [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if content contains any entries that are not of type [XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html)

### newKeyName

public abstract [KeyName](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyName.html) **newKeyName**([String](http://docs.google.com/java/lang/String.html) name)

Creates a KeyName from the specified name.

**Parameters:**name - the name that identifies the key **Returns:**a KeyName **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if name is null

### newKeyValue

public abstract [KeyValue](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyValue.html) **newKeyValue**([PublicKey](http://docs.google.com/java/security/PublicKey.html) key)  
 throws [KeyException](http://docs.google.com/java/security/KeyException.html)

Creates a KeyValue from the specified public key.

**Parameters:**key - the public key **Returns:**a KeyValue **Throws:** [KeyException](http://docs.google.com/java/security/KeyException.html) - if the key's algorithm is not recognized or supported by this KeyInfoFactory [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if key is null

### newPGPData

public abstract [PGPData](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/PGPData.html) **newPGPData**(byte[] keyId)

Creates a PGPData from the specified PGP public key identifier.

**Parameters:**keyId - a PGP public key identifier as defined in [RFC 2440](http://www.ietf.org/rfc/rfc2440.txt), section 11.2. The array is cloned to protect against subsequent modification. **Returns:**a PGPData **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if keyId is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the key id is not in the correct format

### newPGPData

public abstract [PGPData](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/PGPData.html) **newPGPData**(byte[] keyId,  
 byte[] keyPacket,  
 [List](http://docs.google.com/java/util/List.html) other)

Creates a PGPData from the specified PGP public key identifier, and optional key material packet and list of external elements.

**Parameters:**keyId - a PGP public key identifier as defined in [RFC 2440](http://www.ietf.org/rfc/rfc2440.txt), section 11.2. The array is cloned to protect against subsequent modification.keyPacket - a PGP key material packet as defined in [RFC 2440](http://www.ietf.org/rfc/rfc2440.txt), section 5.5. The array is cloned to protect against subsequent modification. May be null.other - a list of [XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html)s representing elements from an external namespace. The list is defensively copied to protect against subsequent modification. May be null or empty. **Returns:**a PGPData **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if keyId is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the keyId or keyPacket is not in the correct format. For keyPacket, the format of the packet header is checked and the tag is verified that it is of type key material. The contents and format of the packet body are not checked. [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if other contains any entries that are not of type [XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html)

### newPGPData

public abstract [PGPData](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/PGPData.html) **newPGPData**(byte[] keyPacket,  
 [List](http://docs.google.com/java/util/List.html) other)

Creates a PGPData from the specified PGP key material packet and optional list of external elements.

**Parameters:**keyPacket - a PGP key material packet as defined in [RFC 2440](http://www.ietf.org/rfc/rfc2440.txt), section 5.5. The array is cloned to protect against subsequent modification.other - a list of [XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html)s representing elements from an external namespace. The list is defensively copied to protect against subsequent modification. May be null or empty. **Returns:**a PGPData **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if keyPacket is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if keyPacket is not in the correct format. For keyPacket, the format of the packet header is checked and the tag is verified that it is of type key material. The contents and format of the packet body are not checked. [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if other contains any entries that are not of type [XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html)

### newRetrievalMethod

public abstract [RetrievalMethod](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/RetrievalMethod.html) **newRetrievalMethod**([String](http://docs.google.com/java/lang/String.html) uri)

Creates a RetrievalMethod from the specified URI.

**Parameters:**uri - the URI that identifies the KeyInfo information to be retrieved **Returns:**a RetrievalMethod **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if uri is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if uri is not RFC 2396 compliant

### newRetrievalMethod

public abstract [RetrievalMethod](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/RetrievalMethod.html) **newRetrievalMethod**([String](http://docs.google.com/java/lang/String.html) uri,  
 [String](http://docs.google.com/java/lang/String.html) type,  
 [List](http://docs.google.com/java/util/List.html) transforms)

Creates a RetrievalMethod from the specified parameters.

**Parameters:**uri - the URI that identifies the KeyInfo information to be retrievedtype - a URI that identifies the type of KeyInfo information to be retrieved (may be null)transforms - a list of [Transform](http://docs.google.com/javax/xml/crypto/dsig/Transform.html)s. The list is defensively copied to protect against subsequent modification. May be null or empty. **Returns:**a RetrievalMethod **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if uri is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if uri is not RFC 2396 compliant [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if transforms contains any entries that are not of type [Transform](http://docs.google.com/javax/xml/crypto/dsig/Transform.html)

### newX509Data

public abstract [X509Data](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/X509Data.html) **newX509Data**([List](http://docs.google.com/java/util/List.html) content)

Creates a X509Data containing the specified list of X.509 content.

**Parameters:**content - a list of one or more X.509 content types. Valid types are [String](http://docs.google.com/java/lang/String.html) (subject names), byte[] (subject key ids), [X509Certificate](http://docs.google.com/java/security/cert/X509Certificate.html), [X509CRL](http://docs.google.com/java/security/cert/X509CRL.html), or [XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html) ([X509IssuerSerial](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/X509IssuerSerial.html) objects or elements from an external namespace). Subject names are distinguished names in RFC 2253 String format. Implementations MUST support the attribute type keywords defined in RFC 2253 (CN, L, ST, O, OU, C, STREET, DC and UID). Implementations MAY support additional keywords. The list is defensively copied to protect against subsequent modification. **Returns:**a X509Data **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if content is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if content is empty, or if a subject name is not RFC 2253 compliant or one of the attribute type keywords is not recognized. [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if content contains any entries that are not of one of the valid types mentioned above

### newX509IssuerSerial

public abstract [X509IssuerSerial](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/X509IssuerSerial.html) **newX509IssuerSerial**([String](http://docs.google.com/java/lang/String.html) issuerName,  
 [BigInteger](http://docs.google.com/java/math/BigInteger.html) serialNumber)

Creates an X509IssuerSerial from the specified X.500 issuer distinguished name and serial number.

**Parameters:**issuerName - the issuer's distinguished name in RFC 2253 String format. Implementations MUST support the attribute type keywords defined in RFC 2253 (CN, L, ST, O, OU, C, STREET, DC and UID). Implementations MAY support additional keywords.serialNumber - the serial number **Returns:**an X509IssuerSerial **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if issuerName or serialNumber are null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the issuer name is not RFC 2253 compliant or one of the attribute type keywords is not recognized.

### isFeatureSupported

public abstract boolean **isFeatureSupported**([String](http://docs.google.com/java/lang/String.html) feature)

Indicates whether a specified feature is supported.

**Parameters:**feature - the feature name (as an absolute URI) **Returns:**true if the specified feature is supported, false otherwise **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if feature is null

### getURIDereferencer

public abstract [URIDereferencer](http://docs.google.com/javax/xml/crypto/URIDereferencer.html) **getURIDereferencer**()

Returns a reference to the URIDereferencer that is used by default to dereference URIs in [RetrievalMethod](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/RetrievalMethod.html) objects.

**Returns:**a reference to the default URIDereferencer

### unmarshalKeyInfo

public abstract [KeyInfo](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfo.html) **unmarshalKeyInfo**([XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html) xmlStructure)  
 throws [MarshalException](http://docs.google.com/javax/xml/crypto/MarshalException.html)

Unmarshals a new KeyInfo instance from a mechanism-specific XMLStructure (ex: [DOMStructure](http://docs.google.com/javax/xml/crypto/dom/DOMStructure.html)) instance.

**Parameters:**xmlStructure - a mechanism-specific XML structure from which to unmarshal the keyinfo from **Returns:**the KeyInfo **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if xmlStructure is null [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the type of xmlStructure is inappropriate for this factory [MarshalException](http://docs.google.com/javax/xml/crypto/MarshalException.html) - if an unrecoverable exception occurs during unmarshalling

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/KeyInfoFactory.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfo.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyName.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html)    [**NO FRAMES**](http://docs.google.com/KeyInfoFactory.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).