| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/XMLSignatureFactory.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/XMLSignatureException.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignContext.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/crypto/dsig/XMLSignatureFactory.html)    [**NO FRAMES**](http://docs.google.com/XMLSignatureFactory.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**

Class XMLSignatureFactory

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

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

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

## XMLSignatureFactory Type

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

XMLSignatureFactory factory = XMLSignatureFactory.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.

XMLSignatureFactory 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("XMLSignatureFactory.DOM", "org.example.DOMXMLSignatureFactory");

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

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

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

## Creating XMLSignatures from scratch

Once the XMLSignatureFactory has been created, objects can be instantiated by calling the appropriate method. For example, a [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html) instance may be created by invoking one of the [newReference](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newReference(java.lang.String,%20javax.xml.crypto.dsig.DigestMethod)) methods.

## Unmarshalling XMLSignatures from XML

Alternatively, an XMLSignature may be created from an existing XML representation by invoking the [unmarshalXMLSignature](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#unmarshalXMLSignature(javax.xml.crypto.dsig.XMLValidateContext)) method and passing it a mechanism-specific [XMLValidateContext](http://docs.google.com/javax/xml/crypto/dsig/XMLValidateContext.html) instance containing the XML content:

DOMValidateContext context = new DOMValidateContext(key, signatureElement);  
 XMLSignature signature = factory.unmarshalXMLSignature(context);

Each XMLSignatureFactory must support the required XMLValidateContext types for that factory type, but may support others. A DOM XMLSignatureFactory must support [DOMValidateContext](http://docs.google.com/javax/xml/crypto/dsig/dom/DOMValidateContext.html) objects.

## Signing and marshalling XMLSignatures to XML

Each XMLSignature created by the factory can also be marshalled to an XML representation and signed, by invoking the [sign](http://docs.google.com/javax/xml/crypto/dsig/XMLSignature.html#sign(javax.xml.crypto.dsig.XMLSignContext)) method of the [XMLSignature](http://docs.google.com/javax/xml/crypto/dsig/XMLSignature.html) object and passing it a mechanism-specific [XMLSignContext](http://docs.google.com/javax/xml/crypto/dsig/XMLSignContext.html) object containing the signing key and marshalling parameters (see [DOMSignContext](http://docs.google.com/javax/xml/crypto/dsig/dom/DOMSignContext.html)). For example:

DOMSignContext context = new DOMSignContext(privateKey, document);  
 signature.sign(context);

**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 XMLSignatureFactory instance concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating a different XMLSignatureFactory instance need not synchronize.

**Since:** 1.6

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

| **Method Summary** | |
| --- | --- |
| static [XMLSignatureFactory](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html) | [**getInstance**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#getInstance())()            Returns an XMLSignatureFactory that supports the default XML processing mechanism and representation type ("DOM"). |
| static [XMLSignatureFactory](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html) | [**getInstance**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#getInstance(java.lang.String))([String](http://docs.google.com/java/lang/String.html) mechanismType)            Returns an XMLSignatureFactory that supports the specified XML processing mechanism and representation type (ex: "DOM"). |
| static [XMLSignatureFactory](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html) | [**getInstance**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.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 an XMLSignatureFactory that supports the requested XML processing mechanism and representation type (ex: "DOM"), as supplied by the specified provider. |
| static [XMLSignatureFactory](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html) | [**getInstance**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.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 an XMLSignatureFactory that supports the requested XML processing mechanism and representation type (ex: "DOM"), as supplied by the specified provider. |
| [KeyInfoFactory](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfoFactory.html) | [**getKeyInfoFactory**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#getKeyInfoFactory())()            Returns a KeyInfoFactory that creates KeyInfo objects. |
| [String](http://docs.google.com/java/lang/String.html) | [**getMechanismType**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#getMechanismType())()            Returns the type of the XML processing mechanism and representation supported by this XMLSignatureFactory (ex: "DOM"). |
| [Provider](http://docs.google.com/java/security/Provider.html) | [**getProvider**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#getProvider())()            Returns the provider of this XMLSignatureFactory. |
| abstract  [URIDereferencer](http://docs.google.com/javax/xml/crypto/URIDereferencer.html) | [**getURIDereferencer**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#getURIDereferencer())()            Returns a reference to the URIDereferencer that is used by default to dereference URIs in [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html) objects. |
| abstract  boolean | [**isFeatureSupported**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#isFeatureSupported(java.lang.String))([String](http://docs.google.com/java/lang/String.html) feature)            Indicates whether a specified feature is supported. |
| abstract  [CanonicalizationMethod](http://docs.google.com/javax/xml/crypto/dsig/CanonicalizationMethod.html) | [**newCanonicalizationMethod**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newCanonicalizationMethod(java.lang.String,%20javax.xml.crypto.dsig.spec.C14NMethodParameterSpec))([String](http://docs.google.com/java/lang/String.html) algorithm, [C14NMethodParameterSpec](http://docs.google.com/javax/xml/crypto/dsig/spec/C14NMethodParameterSpec.html) params)            Creates a CanonicalizationMethod for the specified algorithm URI and parameters. |
| abstract  [CanonicalizationMethod](http://docs.google.com/javax/xml/crypto/dsig/CanonicalizationMethod.html) | [**newCanonicalizationMethod**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newCanonicalizationMethod(java.lang.String,%20javax.xml.crypto.XMLStructure))([String](http://docs.google.com/java/lang/String.html) algorithm, [XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html) params)            Creates a CanonicalizationMethod for the specified algorithm URI and parameters. |
| abstract  [DigestMethod](http://docs.google.com/javax/xml/crypto/dsig/DigestMethod.html) | [**newDigestMethod**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newDigestMethod(java.lang.String,%20javax.xml.crypto.dsig.spec.DigestMethodParameterSpec))([String](http://docs.google.com/java/lang/String.html) algorithm, [DigestMethodParameterSpec](http://docs.google.com/javax/xml/crypto/dsig/spec/DigestMethodParameterSpec.html) params)            Creates a DigestMethod for the specified algorithm URI and parameters. |
| abstract  [Manifest](http://docs.google.com/javax/xml/crypto/dsig/Manifest.html) | [**newManifest**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newManifest(java.util.List))([List](http://docs.google.com/java/util/List.html) references)            Creates a Manifest containing the specified list of [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html)s. |
| abstract  [Manifest](http://docs.google.com/javax/xml/crypto/dsig/Manifest.html) | [**newManifest**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newManifest(java.util.List,%20java.lang.String))([List](http://docs.google.com/java/util/List.html) references, [String](http://docs.google.com/java/lang/String.html) id)            Creates a Manifest containing the specified list of [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html)s and optional id. |
| abstract  [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html) | [**newReference**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newReference(java.lang.String,%20javax.xml.crypto.dsig.DigestMethod))([String](http://docs.google.com/java/lang/String.html) uri, [DigestMethod](http://docs.google.com/javax/xml/crypto/dsig/DigestMethod.html) dm)            Creates a Reference with the specified URI and digest method. |
| abstract  [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html) | [**newReference**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newReference(java.lang.String,%20javax.xml.crypto.dsig.DigestMethod,%20java.util.List,%20javax.xml.crypto.Data,%20java.util.List,%20java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) uri, [DigestMethod](http://docs.google.com/javax/xml/crypto/dsig/DigestMethod.html) dm, [List](http://docs.google.com/java/util/List.html) appliedTransforms, [Data](http://docs.google.com/javax/xml/crypto/Data.html) result, [List](http://docs.google.com/java/util/List.html) transforms, [String](http://docs.google.com/java/lang/String.html) type, [String](http://docs.google.com/java/lang/String.html) id)            Creates a Reference with the specified parameters. |
| abstract  [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html) | [**newReference**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newReference(java.lang.String,%20javax.xml.crypto.dsig.DigestMethod,%20java.util.List,%20java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) uri, [DigestMethod](http://docs.google.com/javax/xml/crypto/dsig/DigestMethod.html) dm, [List](http://docs.google.com/java/util/List.html) transforms, [String](http://docs.google.com/java/lang/String.html) type, [String](http://docs.google.com/java/lang/String.html) id)            Creates a Reference with the specified parameters. |
| abstract  [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html) | [**newReference**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newReference(java.lang.String,%20javax.xml.crypto.dsig.DigestMethod,%20java.util.List,%20java.lang.String,%20java.lang.String,%20byte%5B%5D))([String](http://docs.google.com/java/lang/String.html) uri, [DigestMethod](http://docs.google.com/javax/xml/crypto/dsig/DigestMethod.html) dm, [List](http://docs.google.com/java/util/List.html) transforms, [String](http://docs.google.com/java/lang/String.html) type, [String](http://docs.google.com/java/lang/String.html) id, byte[] digestValue)            Creates a Reference with the specified parameters and pre-calculated digest value. |
| abstract  [SignatureMethod](http://docs.google.com/javax/xml/crypto/dsig/SignatureMethod.html) | [**newSignatureMethod**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newSignatureMethod(java.lang.String,%20javax.xml.crypto.dsig.spec.SignatureMethodParameterSpec))([String](http://docs.google.com/java/lang/String.html) algorithm, [SignatureMethodParameterSpec](http://docs.google.com/javax/xml/crypto/dsig/spec/SignatureMethodParameterSpec.html) params)            Creates a SignatureMethod for the specified algorithm URI and parameters. |
| abstract  [SignatureProperties](http://docs.google.com/javax/xml/crypto/dsig/SignatureProperties.html) | [**newSignatureProperties**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newSignatureProperties(java.util.List,%20java.lang.String))([List](http://docs.google.com/java/util/List.html) properties, [String](http://docs.google.com/java/lang/String.html) id)            Creates a SignatureProperties containing the specified list of [SignatureProperty](http://docs.google.com/javax/xml/crypto/dsig/SignatureProperty.html)s and optional id. |
| abstract  [SignatureProperty](http://docs.google.com/javax/xml/crypto/dsig/SignatureProperty.html) | [**newSignatureProperty**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newSignatureProperty(java.util.List,%20java.lang.String,%20java.lang.String))([List](http://docs.google.com/java/util/List.html) content, [String](http://docs.google.com/java/lang/String.html) target, [String](http://docs.google.com/java/lang/String.html) id)            Creates a SignatureProperty containing the specified list of [XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html)s, target URI and optional id. |
| abstract  [SignedInfo](http://docs.google.com/javax/xml/crypto/dsig/SignedInfo.html) | [**newSignedInfo**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newSignedInfo(javax.xml.crypto.dsig.CanonicalizationMethod,%20javax.xml.crypto.dsig.SignatureMethod,%20java.util.List))([CanonicalizationMethod](http://docs.google.com/javax/xml/crypto/dsig/CanonicalizationMethod.html) cm, [SignatureMethod](http://docs.google.com/javax/xml/crypto/dsig/SignatureMethod.html) sm, [List](http://docs.google.com/java/util/List.html) references)            Creates a SignedInfo with the specified canonicalization and signature methods, and list of one or more references. |
| abstract  [SignedInfo](http://docs.google.com/javax/xml/crypto/dsig/SignedInfo.html) | [**newSignedInfo**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newSignedInfo(javax.xml.crypto.dsig.CanonicalizationMethod,%20javax.xml.crypto.dsig.SignatureMethod,%20java.util.List,%20java.lang.String))([CanonicalizationMethod](http://docs.google.com/javax/xml/crypto/dsig/CanonicalizationMethod.html) cm, [SignatureMethod](http://docs.google.com/javax/xml/crypto/dsig/SignatureMethod.html) sm, [List](http://docs.google.com/java/util/List.html) references, [String](http://docs.google.com/java/lang/String.html) id)            Creates a SignedInfo with the specified parameters. |
| abstract  [Transform](http://docs.google.com/javax/xml/crypto/dsig/Transform.html) | [**newTransform**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newTransform(java.lang.String,%20javax.xml.crypto.dsig.spec.TransformParameterSpec))([String](http://docs.google.com/java/lang/String.html) algorithm, [TransformParameterSpec](http://docs.google.com/javax/xml/crypto/dsig/spec/TransformParameterSpec.html) params)            Creates a Transform for the specified algorithm URI and parameters. |
| abstract  [Transform](http://docs.google.com/javax/xml/crypto/dsig/Transform.html) | [**newTransform**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newTransform(java.lang.String,%20javax.xml.crypto.XMLStructure))([String](http://docs.google.com/java/lang/String.html) algorithm, [XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html) params)            Creates a Transform for the specified algorithm URI and parameters. |
| abstract  [XMLObject](http://docs.google.com/javax/xml/crypto/dsig/XMLObject.html) | [**newXMLObject**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newXMLObject(java.util.List,%20java.lang.String,%20java.lang.String,%20java.lang.String))([List](http://docs.google.com/java/util/List.html) content, [String](http://docs.google.com/java/lang/String.html) id, [String](http://docs.google.com/java/lang/String.html) mimeType, [String](http://docs.google.com/java/lang/String.html) encoding)            Creates an XMLObject from the specified parameters. |
| abstract  [XMLSignature](http://docs.google.com/javax/xml/crypto/dsig/XMLSignature.html) | [**newXMLSignature**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newXMLSignature(javax.xml.crypto.dsig.SignedInfo,%20javax.xml.crypto.dsig.keyinfo.KeyInfo))([SignedInfo](http://docs.google.com/javax/xml/crypto/dsig/SignedInfo.html) si, [KeyInfo](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfo.html) ki)            Creates an XMLSignature and initializes it with the contents of the specified SignedInfo and KeyInfo objects. |
| abstract  [XMLSignature](http://docs.google.com/javax/xml/crypto/dsig/XMLSignature.html) | [**newXMLSignature**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#newXMLSignature(javax.xml.crypto.dsig.SignedInfo,%20javax.xml.crypto.dsig.keyinfo.KeyInfo,%20java.util.List,%20java.lang.String,%20java.lang.String))([SignedInfo](http://docs.google.com/javax/xml/crypto/dsig/SignedInfo.html) si, [KeyInfo](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfo.html) ki, [List](http://docs.google.com/java/util/List.html) objects, [String](http://docs.google.com/java/lang/String.html) id, [String](http://docs.google.com/java/lang/String.html) signatureValueId)            Creates an XMLSignature and initializes it with the specified parameters. |
| abstract  [XMLSignature](http://docs.google.com/javax/xml/crypto/dsig/XMLSignature.html) | [**unmarshalXMLSignature**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#unmarshalXMLSignature(javax.xml.crypto.XMLStructure))([XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html) xmlStructure)            Unmarshals a new XMLSignature instance from a mechanism-specific XMLStructure instance. |
| abstract  [XMLSignature](http://docs.google.com/javax/xml/crypto/dsig/XMLSignature.html) | [**unmarshalXMLSignature**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html#unmarshalXMLSignature(javax.xml.crypto.dsig.XMLValidateContext))([XMLValidateContext](http://docs.google.com/javax/xml/crypto/dsig/XMLValidateContext.html) context)            Unmarshals a new XMLSignature instance from a mechanism-specific XMLValidateContext 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** |
| --- |

### XMLSignatureFactory

protected **XMLSignatureFactory**()

Default constructor, for invocation by subclasses.

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

### getInstance

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

Returns an XMLSignatureFactory 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 an XMLSignatureFactory implementation of the desired mechanism type. It traverses the list of registered security Providers, starting with the most preferred Provider. A new XMLSignatureFactory 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 XMLSignatureFactory **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 an XMLSignatureFactory implementation for the specified mechanism**See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getInstance

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

Returns an XMLSignatureFactory 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 XMLSignatureFactory **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if provider or mechanismType is null [NoSuchMechanismException](http://docs.google.com/javax/xml/crypto/NoSuchMechanismException.html) - if an XMLSignatureFactory 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 [XMLSignatureFactory](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.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 an XMLSignatureFactory 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 XMLSignatureFactory **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 provider or mechanismType is null [NoSuchMechanismException](http://docs.google.com/javax/xml/crypto/NoSuchMechanismException.html) - if an XMLSignatureFactory 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 [XMLSignatureFactory](http://docs.google.com/javax/xml/crypto/dsig/XMLSignatureFactory.html) **getInstance**()

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

This method uses the standard JCA provider lookup mechanism to locate and instantiate an XMLSignatureFactory implementation of the default mechanism type. It traverses the list of registered security Providers, starting with the most preferred Provider. A new XMLSignatureFactory 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 XMLSignatureFactory **Throws:** [NoSuchMechanismException](http://docs.google.com/javax/xml/crypto/NoSuchMechanismException.html) - if no Provider supports an XMLSignatureFactory 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 XMLSignatureFactory (ex: "DOM").

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

### getProvider

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

Returns the provider of this XMLSignatureFactory.

**Returns:**the provider of this XMLSignatureFactory

### newXMLSignature

public abstract [XMLSignature](http://docs.google.com/javax/xml/crypto/dsig/XMLSignature.html) **newXMLSignature**([SignedInfo](http://docs.google.com/javax/xml/crypto/dsig/SignedInfo.html) si,  
 [KeyInfo](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfo.html) ki)

Creates an XMLSignature and initializes it with the contents of the specified SignedInfo and KeyInfo objects.

**Parameters:**si - the signed infoki - the key info (may be null) **Returns:**an XMLSignature **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if si is null

### newXMLSignature

public abstract [XMLSignature](http://docs.google.com/javax/xml/crypto/dsig/XMLSignature.html) **newXMLSignature**([SignedInfo](http://docs.google.com/javax/xml/crypto/dsig/SignedInfo.html) si,  
 [KeyInfo](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfo.html) ki,  
 [List](http://docs.google.com/java/util/List.html) objects,  
 [String](http://docs.google.com/java/lang/String.html) id,  
 [String](http://docs.google.com/java/lang/String.html) signatureValueId)

Creates an XMLSignature and initializes it with the specified parameters.

**Parameters:**si - the signed infoki - the key info (may be null)objects - a list of [XMLObject](http://docs.google.com/javax/xml/crypto/dsig/XMLObject.html)s (may be empty or null)id - the Id (may be null)signatureValueId - the SignatureValue Id (may be null) **Returns:**an XMLSignature **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if si is null [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if any of the objects are not of type XMLObject

### newReference

public abstract [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html) **newReference**([String](http://docs.google.com/java/lang/String.html) uri,  
 [DigestMethod](http://docs.google.com/javax/xml/crypto/dsig/DigestMethod.html) dm)

Creates a Reference with the specified URI and digest method.

**Parameters:**uri - the reference URI (may be null)dm - the digest method **Returns:**a Reference **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if uri is not RFC 2396 compliant [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if dm is null

### newReference

public abstract [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html) **newReference**([String](http://docs.google.com/java/lang/String.html) uri,  
 [DigestMethod](http://docs.google.com/javax/xml/crypto/dsig/DigestMethod.html) dm,  
 [List](http://docs.google.com/java/util/List.html) transforms,  
 [String](http://docs.google.com/java/lang/String.html) type,  
 [String](http://docs.google.com/java/lang/String.html) id)

Creates a Reference with the specified parameters.

**Parameters:**uri - the reference URI (may be null)dm - the digest methodtransforms - 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.type - the reference type, as a URI (may be null)id - the reference ID (may be null) **Returns:**a Reference **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if any of the transforms are not of type Transform [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if uri is not RFC 2396 compliant [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if dm is null

### newReference

public abstract [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html) **newReference**([String](http://docs.google.com/java/lang/String.html) uri,  
 [DigestMethod](http://docs.google.com/javax/xml/crypto/dsig/DigestMethod.html) dm,  
 [List](http://docs.google.com/java/util/List.html) transforms,  
 [String](http://docs.google.com/java/lang/String.html) type,  
 [String](http://docs.google.com/java/lang/String.html) id,  
 byte[] digestValue)

Creates a Reference with the specified parameters and pre-calculated digest value.

This method is useful when the digest value of a Reference has been previously computed. See for example, the  [OASIS-DSS (Digital Signature Services)](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=dss) specification.

**Parameters:**uri - the reference URI (may be null)dm - the digest methodtransforms - 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.type - the reference type, as a URI (may be null)id - the reference ID (may be null)digestValue - the digest value. The array is cloned to protect against subsequent modification. **Returns:**a Reference **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if any of the transforms are not of type Transform [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if uri is not RFC 2396 compliant [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if dm or digestValue is null

### newReference

public abstract [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html) **newReference**([String](http://docs.google.com/java/lang/String.html) uri,  
 [DigestMethod](http://docs.google.com/javax/xml/crypto/dsig/DigestMethod.html) dm,  
 [List](http://docs.google.com/java/util/List.html) appliedTransforms,  
 [Data](http://docs.google.com/javax/xml/crypto/Data.html) result,  
 [List](http://docs.google.com/java/util/List.html) transforms,  
 [String](http://docs.google.com/java/lang/String.html) type,  
 [String](http://docs.google.com/java/lang/String.html) id)

Creates a Reference with the specified parameters.

This method is useful when a list of transforms have already been applied to the Reference. See for example, the  [OASIS-DSS (Digital Signature Services)](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=dss) specification.

When an XMLSignature containing this reference is generated, the specified transforms (if non-null) are applied to the specified result. The Transforms element of the resulting Reference element is set to the concatenation of the appliedTransforms and transforms.

**Parameters:**uri - the reference URI (may be null)dm - the digest methodappliedTransforms - a list of [Transform](http://docs.google.com/javax/xml/crypto/dsig/Transform.html)s that have already been applied. The list is defensively copied to protect against subsequent modification. The list must contain at least one entry.result - the result of processing the sequence of appliedTransformstransforms - a list of [Transform](http://docs.google.com/javax/xml/crypto/dsig/Transform.html)s that are to be applied when generating the signature. The list is defensively copied to protect against subsequent modification. May be null or empty.type - the reference type, as a URI (may be null)id - the reference ID (may be null) **Returns:**a Reference **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if any of the transforms (in either list) are not of type Transform [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if uri is not RFC 2396 compliant or appliedTransforms is empty [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if dm, appliedTransforms or result is null

### newSignedInfo

public abstract [SignedInfo](http://docs.google.com/javax/xml/crypto/dsig/SignedInfo.html) **newSignedInfo**([CanonicalizationMethod](http://docs.google.com/javax/xml/crypto/dsig/CanonicalizationMethod.html) cm,  
 [SignatureMethod](http://docs.google.com/javax/xml/crypto/dsig/SignatureMethod.html) sm,  
 [List](http://docs.google.com/java/util/List.html) references)

Creates a SignedInfo with the specified canonicalization and signature methods, and list of one or more references.

**Parameters:**cm - the canonicalization methodsm - the signature methodreferences - a list of one or more [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html)s. The list is defensively copied to protect against subsequent modification. **Returns:**a SignedInfo **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if any of the references are not of type Reference [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if references is empty [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if any of the parameters are null

### newSignedInfo

public abstract [SignedInfo](http://docs.google.com/javax/xml/crypto/dsig/SignedInfo.html) **newSignedInfo**([CanonicalizationMethod](http://docs.google.com/javax/xml/crypto/dsig/CanonicalizationMethod.html) cm,  
 [SignatureMethod](http://docs.google.com/javax/xml/crypto/dsig/SignatureMethod.html) sm,  
 [List](http://docs.google.com/java/util/List.html) references,  
 [String](http://docs.google.com/java/lang/String.html) id)

Creates a SignedInfo with the specified parameters.

**Parameters:**cm - the canonicalization methodsm - the signature methodreferences - a list of one or more [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html)s. The list is defensively copied to protect against subsequent modification.id - the id (may be null) **Returns:**a SignedInfo **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if any of the references are not of type Reference [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if references is empty [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if cm, sm, or references are null

### newXMLObject

public abstract [XMLObject](http://docs.google.com/javax/xml/crypto/dsig/XMLObject.html) **newXMLObject**([List](http://docs.google.com/java/util/List.html) content,  
 [String](http://docs.google.com/java/lang/String.html) id,  
 [String](http://docs.google.com/java/lang/String.html) mimeType,  
 [String](http://docs.google.com/java/lang/String.html) encoding)

Creates an XMLObject from the specified parameters.

**Parameters:**content - a list of [XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html)s. The list is defensively copied to protect against subsequent modification. May be null or empty.id - the Id (may be null)mimeType - the mime type (may be null)encoding - the encoding (may be null) **Returns:**an XMLObject **Throws:** [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)

### newManifest

public abstract [Manifest](http://docs.google.com/javax/xml/crypto/dsig/Manifest.html) **newManifest**([List](http://docs.google.com/java/util/List.html) references)

Creates a Manifest containing the specified list of [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html)s.

**Parameters:**references - a list of one or more References. The list is defensively copied to protect against subsequent modification. **Returns:**a Manifest **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if references is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if references is empty [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if references contains any entries that are not of type [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html)

### newManifest

public abstract [Manifest](http://docs.google.com/javax/xml/crypto/dsig/Manifest.html) **newManifest**([List](http://docs.google.com/java/util/List.html) references,  
 [String](http://docs.google.com/java/lang/String.html) id)

Creates a Manifest containing the specified list of [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html)s and optional id.

**Parameters:**references - a list of one or more References. The list is defensively copied to protect against subsequent modification.id - the id (may be null) **Returns:**a Manifest **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if references is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if references is empty [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if references contains any entries that are not of type [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html)

### newSignatureProperty

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

Creates a SignatureProperty containing the specified list of [XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html)s, target URI and optional id.

**Parameters:**content - a list of one or more XMLStructures. The list is defensively copied to protect against subsequent modification.target - the target URI of the Signature that this property applies toid - the id (may be null) **Returns:**a SignatureProperty **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if content or target 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)

### newSignatureProperties

public abstract [SignatureProperties](http://docs.google.com/javax/xml/crypto/dsig/SignatureProperties.html) **newSignatureProperties**([List](http://docs.google.com/java/util/List.html) properties,  
 [String](http://docs.google.com/java/lang/String.html) id)

Creates a SignatureProperties containing the specified list of [SignatureProperty](http://docs.google.com/javax/xml/crypto/dsig/SignatureProperty.html)s and optional id.

**Parameters:**properties - a list of one or more SignaturePropertys. The list is defensively copied to protect against subsequent modification.id - the id (may be null) **Returns:**a SignatureProperties **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if properties is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if properties is empty [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if properties contains any entries that are not of type [SignatureProperty](http://docs.google.com/javax/xml/crypto/dsig/SignatureProperty.html)

### newDigestMethod

public abstract [DigestMethod](http://docs.google.com/javax/xml/crypto/dsig/DigestMethod.html) **newDigestMethod**([String](http://docs.google.com/java/lang/String.html) algorithm,  
 [DigestMethodParameterSpec](http://docs.google.com/javax/xml/crypto/dsig/spec/DigestMethodParameterSpec.html) params)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html),  
 [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html)

Creates a DigestMethod for the specified algorithm URI and parameters.

**Parameters:**algorithm - the URI identifying the digest algorithmparams - algorithm-specific digest parameters (may be null) **Returns:**the DigestMethod **Throws:** [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if the specified parameters are inappropriate for the requested algorithm [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if an implementation of the specified algorithm cannot be found [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if algorithm is null

### newSignatureMethod

public abstract [SignatureMethod](http://docs.google.com/javax/xml/crypto/dsig/SignatureMethod.html) **newSignatureMethod**([String](http://docs.google.com/java/lang/String.html) algorithm,  
 [SignatureMethodParameterSpec](http://docs.google.com/javax/xml/crypto/dsig/spec/SignatureMethodParameterSpec.html) params)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html),  
 [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html)

Creates a SignatureMethod for the specified algorithm URI and parameters.

**Parameters:**algorithm - the URI identifying the signature algorithmparams - algorithm-specific signature parameters (may be null) **Returns:**the SignatureMethod **Throws:** [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if the specified parameters are inappropriate for the requested algorithm [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if an implementation of the specified algorithm cannot be found [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if algorithm is null

### newTransform

public abstract [Transform](http://docs.google.com/javax/xml/crypto/dsig/Transform.html) **newTransform**([String](http://docs.google.com/java/lang/String.html) algorithm,  
 [TransformParameterSpec](http://docs.google.com/javax/xml/crypto/dsig/spec/TransformParameterSpec.html) params)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html),  
 [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html)

Creates a Transform for the specified algorithm URI and parameters.

**Parameters:**algorithm - the URI identifying the transform algorithmparams - algorithm-specific transform parameters (may be null) **Returns:**the Transform **Throws:** [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if the specified parameters are inappropriate for the requested algorithm [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if an implementation of the specified algorithm cannot be found [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if algorithm is null

### newTransform

public abstract [Transform](http://docs.google.com/javax/xml/crypto/dsig/Transform.html) **newTransform**([String](http://docs.google.com/java/lang/String.html) algorithm,  
 [XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html) params)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html),  
 [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html)

Creates a Transform for the specified algorithm URI and parameters. The parameters are specified as a mechanism-specific XMLStructure (ex: [DOMStructure](http://docs.google.com/javax/xml/crypto/dom/DOMStructure.html)). This method is useful when the parameters are in XML form or there is no standard class for specifying the parameters.

**Parameters:**algorithm - the URI identifying the transform algorithmparams - a mechanism-specific XML structure from which to unmarshal the parameters from (may be null if not required or optional) **Returns:**the Transform **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the type of params is inappropriate for this XMLSignatureFactory [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if the specified parameters are inappropriate for the requested algorithm [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if an implementation of the specified algorithm cannot be found [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if algorithm is null

### newCanonicalizationMethod

public abstract [CanonicalizationMethod](http://docs.google.com/javax/xml/crypto/dsig/CanonicalizationMethod.html) **newCanonicalizationMethod**([String](http://docs.google.com/java/lang/String.html) algorithm,  
 [C14NMethodParameterSpec](http://docs.google.com/javax/xml/crypto/dsig/spec/C14NMethodParameterSpec.html) params)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html),  
 [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html)

Creates a CanonicalizationMethod for the specified algorithm URI and parameters.

**Parameters:**algorithm - the URI identifying the canonicalization algorithmparams - algorithm-specific canonicalization parameters (may be null) **Returns:**the CanonicalizationMethod **Throws:** [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if the specified parameters are inappropriate for the requested algorithm [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if an implementation of the specified algorithm cannot be found [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if algorithm is null

### newCanonicalizationMethod

public abstract [CanonicalizationMethod](http://docs.google.com/javax/xml/crypto/dsig/CanonicalizationMethod.html) **newCanonicalizationMethod**([String](http://docs.google.com/java/lang/String.html) algorithm,  
 [XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html) params)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html),  
 [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html)

Creates a CanonicalizationMethod for the specified algorithm URI and parameters. The parameters are specified as a mechanism-specific XMLStructure (ex: [DOMStructure](http://docs.google.com/javax/xml/crypto/dom/DOMStructure.html)). This method is useful when the parameters are in XML form or there is no standard class for specifying the parameters.

**Parameters:**algorithm - the URI identifying the canonicalization algorithmparams - a mechanism-specific XML structure from which to unmarshal the parameters from (may be null if not required or optional) **Returns:**the CanonicalizationMethod **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the type of params is inappropriate for this XMLSignatureFactory [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if the specified parameters are inappropriate for the requested algorithm [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if an implementation of the specified algorithm cannot be found [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if algorithm is null

### getKeyInfoFactory

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

Returns a KeyInfoFactory that creates KeyInfo objects. The returned KeyInfoFactory has the same mechanism type and provider as this XMLSignatureFactory.

**Returns:**a KeyInfoFactory **Throws:** [NoSuchMechanismException](http://docs.google.com/javax/xml/crypto/NoSuchMechanismException.html) - if a KeyFactory implementation with the same mechanism type and provider is not available

### unmarshalXMLSignature

public abstract [XMLSignature](http://docs.google.com/javax/xml/crypto/dsig/XMLSignature.html) **unmarshalXMLSignature**([XMLValidateContext](http://docs.google.com/javax/xml/crypto/dsig/XMLValidateContext.html) context)  
 throws [MarshalException](http://docs.google.com/javax/xml/crypto/MarshalException.html)

Unmarshals a new XMLSignature instance from a mechanism-specific XMLValidateContext instance.

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

### unmarshalXMLSignature

public abstract [XMLSignature](http://docs.google.com/javax/xml/crypto/dsig/XMLSignature.html) **unmarshalXMLSignature**([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 XMLSignature instance from a mechanism-specific XMLStructure instance. This method is useful if you only want to unmarshal (and not validate) an XMLSignature.

**Parameters:**xmlStructure - a mechanism-specific XML structure from which to unmarshal the signature from **Returns:**the XMLSignature **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

### 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 [Reference](http://docs.google.com/javax/xml/crypto/dsig/Reference.html) objects.

**Returns:**a reference to the default URIDereferencer (never null)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/XMLSignatureFactory.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/XMLSignatureException.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/crypto/dsig/XMLSignContext.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/crypto/dsig/XMLSignatureFactory.html)    [**NO FRAMES**](http://docs.google.com/XMLSignatureFactory.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).