| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TransformService.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/TransformException.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/crypto/dsig/XMLObject.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/crypto/dsig/TransformService.html)    [**NO FRAMES**](http://docs.google.com/TransformService.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

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

Class TransformService

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

**All Implemented Interfaces:** [AlgorithmMethod](http://docs.google.com/javax/xml/crypto/AlgorithmMethod.html), [Transform](http://docs.google.com/javax/xml/crypto/dsig/Transform.html), [XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html)

public abstract class **TransformService**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Transform](http://docs.google.com/javax/xml/crypto/dsig/Transform.html)

A Service Provider Interface for transform and canonicalization algorithms.

Each instance of TransformService supports a specific transform or canonicalization algorithm and XML mechanism type. To create a TransformService, call one of the static [getInstance](http://docs.google.com/javax/xml/crypto/dsig/TransformService.html#getInstance(java.lang.String,%20java.lang.String)) methods, passing in the algorithm URI and XML mechanism type desired, for example:

TransformService ts = TransformService.getInstance(Transform.XPATH2, "DOM");

TransformService implementations are registered and loaded using the [Provider](http://docs.google.com/java/security/Provider.html) mechanism. Each TransformService service provider implementation should include a MechanismType service attribute that identifies the XML mechanism type that it supports. If the attribute is not specified, "DOM" is assumed. For example, a service provider that supports the XPath Filter 2 Transform and DOM mechanism would be specified in the Provider subclass as:

put("TransformService." + Transform.XPATH2,  
 "org.example.XPath2TransformService");  
 put("TransformService." + Transform.XPATH2 + " MechanismType", "DOM");

TransformService implementations that support the DOM mechanism type must abide by the DOM interoperability requirements 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.

Once a TransformService has been created, it can be used to process Transform or CanonicalizationMethod objects. If the Transform or CanonicalizationMethod exists in XML form (for example, when validating an existing XMLSignature), the [init(XMLStructure, XMLCryptoContext)](http://docs.google.com/javax/xml/crypto/dsig/TransformService.html#init(javax.xml.crypto.XMLStructure,%20javax.xml.crypto.XMLCryptoContext)) method must be first called to initialize the transform and provide document context (even if there are no parameters). Alternatively, if the Transform or CanonicalizationMethod is being created from scratch, the [init(TransformParameterSpec)](http://docs.google.com/javax/xml/crypto/dsig/TransformService.html#init(javax.xml.crypto.dsig.spec.TransformParameterSpec)) method is called to initialize the transform with parameters and the [marshalParams](http://docs.google.com/javax/xml/crypto/dsig/TransformService.html#marshalParams(javax.xml.crypto.XMLStructure,%20javax.xml.crypto.XMLCryptoContext)) method is called to marshal the parameters to XML and provide the transform with document context. Finally, the [transform](http://docs.google.com/javax/xml/crypto/dsig/Transform.html#transform(javax.xml.crypto.Data,%20javax.xml.crypto.XMLCryptoContext)) method is called to perform the transformation.

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

**Since:** 1.6

| **Field Summary** | |
| --- | --- |

| **Fields inherited from interface javax.xml.crypto.dsig.**[**Transform**](http://docs.google.com/javax/xml/crypto/dsig/Transform.html) |
| --- |
| [BASE64](http://docs.google.com/javax/xml/crypto/dsig/Transform.html#BASE64), [ENVELOPED](http://docs.google.com/javax/xml/crypto/dsig/Transform.html#ENVELOPED), [XPATH](http://docs.google.com/javax/xml/crypto/dsig/Transform.html#XPATH), [XPATH2](http://docs.google.com/javax/xml/crypto/dsig/Transform.html#XPATH2), [XSLT](http://docs.google.com/javax/xml/crypto/dsig/Transform.html#XSLT) |

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

| **Method Summary** | |
| --- | --- |
| [String](http://docs.google.com/java/lang/String.html) | [**getAlgorithm**](http://docs.google.com/javax/xml/crypto/dsig/TransformService.html#getAlgorithm())()            Returns the URI of the algorithm supported by this TransformService. |
| static [TransformService](http://docs.google.com/javax/xml/crypto/dsig/TransformService.html) | [**getInstance**](http://docs.google.com/javax/xml/crypto/dsig/TransformService.html#getInstance(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) algorithm, [String](http://docs.google.com/java/lang/String.html) mechanismType)            Returns a TransformService that supports the specified algorithm URI (ex: [Transform.XPATH2](http://docs.google.com/javax/xml/crypto/dsig/Transform.html#XPATH2)) and mechanism type (ex: DOM). |
| static [TransformService](http://docs.google.com/javax/xml/crypto/dsig/TransformService.html) | [**getInstance**](http://docs.google.com/javax/xml/crypto/dsig/TransformService.html#getInstance(java.lang.String,%20java.lang.String,%20java.security.Provider))([String](http://docs.google.com/java/lang/String.html) algorithm, [String](http://docs.google.com/java/lang/String.html) mechanismType, [Provider](http://docs.google.com/java/security/Provider.html) provider)            Returns a TransformService that supports the specified algorithm URI (ex: [Transform.XPATH2](http://docs.google.com/javax/xml/crypto/dsig/Transform.html#XPATH2)) and mechanism type (ex: DOM) as supplied by the specified provider. |
| static [TransformService](http://docs.google.com/javax/xml/crypto/dsig/TransformService.html) | [**getInstance**](http://docs.google.com/javax/xml/crypto/dsig/TransformService.html#getInstance(java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) algorithm, [String](http://docs.google.com/java/lang/String.html) mechanismType, [String](http://docs.google.com/java/lang/String.html) provider)            Returns a TransformService that supports the specified algorithm URI (ex: [Transform.XPATH2](http://docs.google.com/javax/xml/crypto/dsig/Transform.html#XPATH2)) and mechanism 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/TransformService.html#getMechanismType())()            Returns the mechanism type supported by this TransformService. |
| [Provider](http://docs.google.com/java/security/Provider.html) | [**getProvider**](http://docs.google.com/javax/xml/crypto/dsig/TransformService.html#getProvider())()            Returns the provider of this TransformService. |
| abstract  void | [**init**](http://docs.google.com/javax/xml/crypto/dsig/TransformService.html#init(javax.xml.crypto.dsig.spec.TransformParameterSpec))([TransformParameterSpec](http://docs.google.com/javax/xml/crypto/dsig/spec/TransformParameterSpec.html) params)            Initializes this TransformService with the specified parameters. |
| abstract  void | [**init**](http://docs.google.com/javax/xml/crypto/dsig/TransformService.html#init(javax.xml.crypto.XMLStructure,%20javax.xml.crypto.XMLCryptoContext))([XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html) parent, [XMLCryptoContext](http://docs.google.com/javax/xml/crypto/XMLCryptoContext.html) context)            Initializes this TransformService with the specified parameters and document context. |
| abstract  void | [**marshalParams**](http://docs.google.com/javax/xml/crypto/dsig/TransformService.html#marshalParams(javax.xml.crypto.XMLStructure,%20javax.xml.crypto.XMLCryptoContext))([XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html) parent, [XMLCryptoContext](http://docs.google.com/javax/xml/crypto/XMLCryptoContext.html) context)            Marshals the algorithm-specific parameters. |

| **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)) |

| **Methods inherited from interface javax.xml.crypto.dsig.**[**Transform**](http://docs.google.com/javax/xml/crypto/dsig/Transform.html) |
| --- |
| [getParameterSpec](http://docs.google.com/javax/xml/crypto/dsig/Transform.html#getParameterSpec()), [transform](http://docs.google.com/javax/xml/crypto/dsig/Transform.html#transform(javax.xml.crypto.Data,%20javax.xml.crypto.XMLCryptoContext)), [transform](http://docs.google.com/javax/xml/crypto/dsig/Transform.html#transform(javax.xml.crypto.Data,%20javax.xml.crypto.XMLCryptoContext,%20java.io.OutputStream)) |

| **Methods inherited from interface javax.xml.crypto.**[**XMLStructure**](http://docs.google.com/javax/xml/crypto/XMLStructure.html) |
| --- |
| [isFeatureSupported](http://docs.google.com/javax/xml/crypto/XMLStructure.html#isFeatureSupported(java.lang.String)) |

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

### TransformService

protected **TransformService**()

Default constructor, for invocation by subclasses.

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

### getInstance

public static [TransformService](http://docs.google.com/javax/xml/crypto/dsig/TransformService.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) algorithm,  
 [String](http://docs.google.com/java/lang/String.html) mechanismType)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html)

Returns a TransformService that supports the specified algorithm URI (ex: [Transform.XPATH2](http://docs.google.com/javax/xml/crypto/dsig/Transform.html#XPATH2)) and mechanism type (ex: DOM).

This method uses the standard JCA provider lookup mechanism to locate and instantiate a TransformService implementation of the desired algorithm and MechanismType service attribute. It traverses the list of registered security Providers, starting with the most preferred Provider. A new TransformService object from the first Provider that supports the specified algorithm and mechanism type 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:**algorithm - the URI of the algorithmmechanismType - the type of the XML processing mechanism and representation **Returns:**a new TransformService **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if algorithm or mechanismType is null [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if no Provider supports a TransformService implementation for the specified algorithm and mechanism type**See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getInstance

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

Returns a TransformService that supports the specified algorithm URI (ex: [Transform.XPATH2](http://docs.google.com/javax/xml/crypto/dsig/Transform.html#XPATH2)) and mechanism 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:**algorithm - the URI of the algorithmmechanismType - the type of the XML processing mechanism and representationprovider - the Provider object **Returns:**a new TransformService **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if provider, algorithm, or mechanismType is null [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if a TransformService implementation for the specified algorithm and mechanism type is not available from the specified Provider object**See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getInstance

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

Returns a TransformService that supports the specified algorithm URI (ex: [Transform.XPATH2](http://docs.google.com/javax/xml/crypto/dsig/Transform.html#XPATH2)) and mechanism 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:**algorithm - the URI of the algorithmmechanismType - the type of the XML processing mechanism and representationprovider - the string name of the provider **Returns:**a new TransformService **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, mechanismType, or algorithm is null [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if a TransformService implementation for the specified algorithm and mechanism type is not available from the specified provider**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 mechanism type supported by this TransformService.

**Returns:**the mechanism type

### getAlgorithm

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

Returns the URI of the algorithm supported by this TransformService.

**Specified by:**[getAlgorithm](http://docs.google.com/javax/xml/crypto/AlgorithmMethod.html#getAlgorithm()) in interface [AlgorithmMethod](http://docs.google.com/javax/xml/crypto/AlgorithmMethod.html) **Returns:**the algorithm URI

### getProvider

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

Returns the provider of this TransformService.

**Returns:**the provider

### init

public abstract void **init**([TransformParameterSpec](http://docs.google.com/javax/xml/crypto/dsig/spec/TransformParameterSpec.html) params)  
 throws [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html)

Initializes this TransformService with the specified parameters.

If the parameters exist in XML form, the [init(XMLStructure, XMLCryptoContext)](http://docs.google.com/javax/xml/crypto/dsig/TransformService.html#init(javax.xml.crypto.XMLStructure,%20javax.xml.crypto.XMLCryptoContext)) method should be used to initialize the TransformService.

**Parameters:**params - the algorithm parameters (may be null if not required or optional) **Throws:** [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if the specified parameters are invalid for this algorithm

### marshalParams

public abstract void **marshalParams**([XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html) parent,  
 [XMLCryptoContext](http://docs.google.com/javax/xml/crypto/XMLCryptoContext.html) context)  
 throws [MarshalException](http://docs.google.com/javax/xml/crypto/MarshalException.html)

Marshals the algorithm-specific parameters. If there are no parameters to be marshalled, this method returns without throwing an exception.

**Parameters:**parent - a mechanism-specific structure containing the parent node that the marshalled parameters should be appended tocontext - the XMLCryptoContext containing additional context (may be null if not applicable) **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the type of parent or context is not compatible with this TransformService [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if parent is null [MarshalException](http://docs.google.com/javax/xml/crypto/MarshalException.html) - if the parameters cannot be marshalled

### init

public abstract void **init**([XMLStructure](http://docs.google.com/javax/xml/crypto/XMLStructure.html) parent,  
 [XMLCryptoContext](http://docs.google.com/javax/xml/crypto/XMLCryptoContext.html) context)  
 throws [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html)

Initializes this TransformService with the specified parameters and document context.

**Parameters:**parent - a mechanism-specific structure containing the parent structurecontext - the XMLCryptoContext containing additional context (may be null if not applicable) **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the type of parent or context is not compatible with this TransformService [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if parent is null [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if the specified parameters are invalid for this algorithm

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TransformService.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/TransformException.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/crypto/dsig/XMLObject.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/crypto/dsig/TransformService.html)    [**NO FRAMES**](http://docs.google.com/TransformService.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

[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).