| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PKIXParameters.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/java/security/cert/PKIXCertPathValidatorResult.html)   [**NEXT CLASS**](http://docs.google.com/java/security/cert/PolicyNode.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/cert/PKIXParameters.html)    [**NO FRAMES**](http://docs.google.com/PKIXParameters.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

## **java.security.cert**

Class PKIXParameters

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.security.cert.PKIXParameters**

**All Implemented Interfaces:** [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [CertPathParameters](http://docs.google.com/java/security/cert/CertPathParameters.html) **Direct Known Subclasses:** [PKIXBuilderParameters](http://docs.google.com/java/security/cert/PKIXBuilderParameters.html)

public class **PKIXParameters**extends [Object](http://docs.google.com/java/lang/Object.html)implements [CertPathParameters](http://docs.google.com/java/security/cert/CertPathParameters.html)

Parameters used as input for the PKIX CertPathValidator algorithm.

A PKIX CertPathValidator uses these parameters to validate a CertPath according to the PKIX certification path validation algorithm.

To instantiate a PKIXParameters object, an application must specify one or more *most-trusted CAs* as defined by the PKIX certification path validation algorithm. The most-trusted CAs can be specified using one of two constructors. An application can call [PKIXParameters(Set)](http://docs.google.com/java/security/cert/PKIXParameters.html#PKIXParameters(java.util.Set)), specifying a Set of TrustAnchor objects, each of which identify a most-trusted CA. Alternatively, an application can call [PKIXParameters(KeyStore)](http://docs.google.com/java/security/cert/PKIXParameters.html#PKIXParameters(java.security.KeyStore)), specifying a KeyStore instance containing trusted certificate entries, each of which will be considered as a most-trusted CA.

Once a PKIXParameters object has been created, other parameters can be specified (by calling [setInitialPolicies](http://docs.google.com/java/security/cert/PKIXParameters.html#setInitialPolicies(java.util.Set)) or [setDate](http://docs.google.com/java/security/cert/PKIXParameters.html#setDate(java.util.Date)), for instance) and then the PKIXParameters is passed along with the CertPath to be validated to [CertPathValidator.validate](http://docs.google.com/java/security/cert/CertPathValidator.html#validate(java.security.cert.CertPath,%20java.security.cert.CertPathParameters)).

Any parameter that is not set (or is set to null) will be set to the default value for that parameter. The default value for the date parameter is null, which indicates the current time when the path is validated. The default for the remaining parameters is the least constrained.

**Concurrent Access**

Unless otherwise specified, the methods defined in this class are not thread-safe. Multiple threads that need to access a single object concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating separate objects need not synchronize.

**Since:** 1.4 **See Also:**[CertPathValidator](http://docs.google.com/java/security/cert/CertPathValidator.html)

| **Constructor Summary** | |
| --- | --- |
| [**PKIXParameters**](http://docs.google.com/java/security/cert/PKIXParameters.html#PKIXParameters(java.security.KeyStore))([KeyStore](http://docs.google.com/java/security/KeyStore.html) keystore)            Creates an instance of PKIXParameters that populates the set of most-trusted CAs from the trusted certificate entries contained in the specified KeyStore. |
| [**PKIXParameters**](http://docs.google.com/java/security/cert/PKIXParameters.html#PKIXParameters(java.util.Set))([Set](http://docs.google.com/java/util/Set.html)<[TrustAnchor](http://docs.google.com/java/security/cert/TrustAnchor.html)> trustAnchors)            Creates an instance of PKIXParameters with the specified Set of most-trusted CAs. |

| **Method Summary** | |
| --- | --- |
| void | [**addCertPathChecker**](http://docs.google.com/java/security/cert/PKIXParameters.html#addCertPathChecker(java.security.cert.PKIXCertPathChecker))([PKIXCertPathChecker](http://docs.google.com/java/security/cert/PKIXCertPathChecker.html) checker)            Adds a PKIXCertPathChecker to the list of certification path checkers. |
| void | [**addCertStore**](http://docs.google.com/java/security/cert/PKIXParameters.html#addCertStore(java.security.cert.CertStore))([CertStore](http://docs.google.com/java/security/cert/CertStore.html) store)            Adds a CertStore to the end of the list of CertStores used in finding certificates and CRLs. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/java/security/cert/PKIXParameters.html#clone())()            Makes a copy of this PKIXParameters object. |
| [List](http://docs.google.com/java/util/List.html)<[PKIXCertPathChecker](http://docs.google.com/java/security/cert/PKIXCertPathChecker.html)> | [**getCertPathCheckers**](http://docs.google.com/java/security/cert/PKIXParameters.html#getCertPathCheckers())()            Returns the List of certification path checkers. |
| [List](http://docs.google.com/java/util/List.html)<[CertStore](http://docs.google.com/java/security/cert/CertStore.html)> | [**getCertStores**](http://docs.google.com/java/security/cert/PKIXParameters.html#getCertStores())()            Returns an immutable List of CertStores that are used to find certificates and CRLs. |
| [Date](http://docs.google.com/java/util/Date.html) | [**getDate**](http://docs.google.com/java/security/cert/PKIXParameters.html#getDate())()            Returns the time for which the validity of the certification path should be determined. |
| [Set](http://docs.google.com/java/util/Set.html)<[String](http://docs.google.com/java/lang/String.html)> | [**getInitialPolicies**](http://docs.google.com/java/security/cert/PKIXParameters.html#getInitialPolicies())()            Returns an immutable Set of initial policy identifiers (OID strings), indicating that any one of these policies would be acceptable to the certificate user for the purposes of certification path processing. |
| boolean | [**getPolicyQualifiersRejected**](http://docs.google.com/java/security/cert/PKIXParameters.html#getPolicyQualifiersRejected())()            Gets the PolicyQualifiersRejected flag. |
| [String](http://docs.google.com/java/lang/String.html) | [**getSigProvider**](http://docs.google.com/java/security/cert/PKIXParameters.html#getSigProvider())()            Returns the signature provider's name, or null if not set. |
| [CertSelector](http://docs.google.com/java/security/cert/CertSelector.html) | [**getTargetCertConstraints**](http://docs.google.com/java/security/cert/PKIXParameters.html#getTargetCertConstraints())()            Returns the required constraints on the target certificate. |
| [Set](http://docs.google.com/java/util/Set.html)<[TrustAnchor](http://docs.google.com/java/security/cert/TrustAnchor.html)> | [**getTrustAnchors**](http://docs.google.com/java/security/cert/PKIXParameters.html#getTrustAnchors())()            Returns an immutable Set of the most-trusted CAs. |
| boolean | [**isAnyPolicyInhibited**](http://docs.google.com/java/security/cert/PKIXParameters.html#isAnyPolicyInhibited())()            Checks whether the any policy OID should be processed if it is included in a certificate. |
| boolean | [**isExplicitPolicyRequired**](http://docs.google.com/java/security/cert/PKIXParameters.html#isExplicitPolicyRequired())()            Checks if explicit policy is required. |
| boolean | [**isPolicyMappingInhibited**](http://docs.google.com/java/security/cert/PKIXParameters.html#isPolicyMappingInhibited())()            Checks if policy mapping is inhibited. |
| boolean | [**isRevocationEnabled**](http://docs.google.com/java/security/cert/PKIXParameters.html#isRevocationEnabled())()            Checks the RevocationEnabled flag. |
| void | [**setAnyPolicyInhibited**](http://docs.google.com/java/security/cert/PKIXParameters.html#setAnyPolicyInhibited(boolean))(boolean val)            Sets state to determine if the any policy OID should be processed if it is included in a certificate. |
| void | [**setCertPathCheckers**](http://docs.google.com/java/security/cert/PKIXParameters.html#setCertPathCheckers(java.util.List))([List](http://docs.google.com/java/util/List.html)<[PKIXCertPathChecker](http://docs.google.com/java/security/cert/PKIXCertPathChecker.html)> checkers)            Sets a List of additional certification path checkers. |
| void | [**setCertStores**](http://docs.google.com/java/security/cert/PKIXParameters.html#setCertStores(java.util.List))([List](http://docs.google.com/java/util/List.html)<[CertStore](http://docs.google.com/java/security/cert/CertStore.html)> stores)            Sets the list of CertStores to be used in finding certificates and CRLs. |
| void | [**setDate**](http://docs.google.com/java/security/cert/PKIXParameters.html#setDate(java.util.Date))([Date](http://docs.google.com/java/util/Date.html) date)            Sets the time for which the validity of the certification path should be determined. |
| void | [**setExplicitPolicyRequired**](http://docs.google.com/java/security/cert/PKIXParameters.html#setExplicitPolicyRequired(boolean))(boolean val)            Sets the ExplicitPolicyRequired flag. |
| void | [**setInitialPolicies**](http://docs.google.com/java/security/cert/PKIXParameters.html#setInitialPolicies(java.util.Set))([Set](http://docs.google.com/java/util/Set.html)<[String](http://docs.google.com/java/lang/String.html)> initialPolicies)            Sets the Set of initial policy identifiers (OID strings), indicating that any one of these policies would be acceptable to the certificate user for the purposes of certification path processing. |
| void | [**setPolicyMappingInhibited**](http://docs.google.com/java/security/cert/PKIXParameters.html#setPolicyMappingInhibited(boolean))(boolean val)            Sets the PolicyMappingInhibited flag. |
| void | [**setPolicyQualifiersRejected**](http://docs.google.com/java/security/cert/PKIXParameters.html#setPolicyQualifiersRejected(boolean))(boolean qualifiersRejected)            Sets the PolicyQualifiersRejected flag. |
| void | [**setRevocationEnabled**](http://docs.google.com/java/security/cert/PKIXParameters.html#setRevocationEnabled(boolean))(boolean val)            Sets the RevocationEnabled flag. |
| void | [**setSigProvider**](http://docs.google.com/java/security/cert/PKIXParameters.html#setSigProvider(java.lang.String))([String](http://docs.google.com/java/lang/String.html) sigProvider)            Sets the signature provider's name. |
| void | [**setTargetCertConstraints**](http://docs.google.com/java/security/cert/PKIXParameters.html#setTargetCertConstraints(java.security.cert.CertSelector))([CertSelector](http://docs.google.com/java/security/cert/CertSelector.html) selector)            Sets the required constraints on the target certificate. |
| void | [**setTrustAnchors**](http://docs.google.com/java/security/cert/PKIXParameters.html#setTrustAnchors(java.util.Set))([Set](http://docs.google.com/java/util/Set.html)<[TrustAnchor](http://docs.google.com/java/security/cert/TrustAnchor.html)> trustAnchors)            Sets the Set of most-trusted CAs. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/security/cert/PKIXParameters.html#toString())()            Returns a formatted string describing the parameters. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [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()), [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** |
| --- |

### PKIXParameters

public **PKIXParameters**([Set](http://docs.google.com/java/util/Set.html)<[TrustAnchor](http://docs.google.com/java/security/cert/TrustAnchor.html)> trustAnchors)  
 throws [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html)

Creates an instance of PKIXParameters with the specified Set of most-trusted CAs. Each element of the set is a [TrustAnchor](http://docs.google.com/java/security/cert/TrustAnchor.html).

Note that the Set is copied to protect against subsequent modifications.

**Parameters:**trustAnchors - a Set of TrustAnchors **Throws:** [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if the specified Set is empty (trustAnchors.isEmpty() == true) [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified Set is null [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if any of the elements in the Set are not of type java.security.cert.TrustAnchor

### PKIXParameters

public **PKIXParameters**([KeyStore](http://docs.google.com/java/security/KeyStore.html) keystore)  
 throws [KeyStoreException](http://docs.google.com/java/security/KeyStoreException.html),  
 [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html)

Creates an instance of PKIXParameters that populates the set of most-trusted CAs from the trusted certificate entries contained in the specified KeyStore. Only keystore entries that contain trusted X509Certificates are considered; all other certificate types are ignored.

**Parameters:**keystore - a KeyStore from which the set of most-trusted CAs will be populated **Throws:** [KeyStoreException](http://docs.google.com/java/security/KeyStoreException.html) - if the keystore has not been initialized [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if the keystore does not contain at least one trusted certificate entry [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the keystore is null

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

### getTrustAnchors

public [Set](http://docs.google.com/java/util/Set.html)<[TrustAnchor](http://docs.google.com/java/security/cert/TrustAnchor.html)> **getTrustAnchors**()

Returns an immutable Set of the most-trusted CAs.

**Returns:**an immutable Set of TrustAnchors (never null)**See Also:**[setTrustAnchors(java.util.Set)](http://docs.google.com/java/security/cert/PKIXParameters.html#setTrustAnchors(java.util.Set))

### setTrustAnchors

public void **setTrustAnchors**([Set](http://docs.google.com/java/util/Set.html)<[TrustAnchor](http://docs.google.com/java/security/cert/TrustAnchor.html)> trustAnchors)  
 throws [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html)

Sets the Set of most-trusted CAs.

Note that the Set is copied to protect against subsequent modifications.

**Parameters:**trustAnchors - a Set of TrustAnchors **Throws:** [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if the specified Set is empty (trustAnchors.isEmpty() == true) [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified Set is null [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if any of the elements in the set are not of type java.security.cert.TrustAnchor**See Also:**[getTrustAnchors()](http://docs.google.com/java/security/cert/PKIXParameters.html#getTrustAnchors())

### getInitialPolicies

public [Set](http://docs.google.com/java/util/Set.html)<[String](http://docs.google.com/java/lang/String.html)> **getInitialPolicies**()

Returns an immutable Set of initial policy identifiers (OID strings), indicating that any one of these policies would be acceptable to the certificate user for the purposes of certification path processing. The default return value is an empty Set, which is interpreted as meaning that any policy would be acceptable.

**Returns:**an immutable Set of initial policy OIDs in String format, or an empty Set (implying any policy is acceptable). Never returns null.**See Also:**[setInitialPolicies(java.util.Set)](http://docs.google.com/java/security/cert/PKIXParameters.html#setInitialPolicies(java.util.Set))

### setInitialPolicies

public void **setInitialPolicies**([Set](http://docs.google.com/java/util/Set.html)<[String](http://docs.google.com/java/lang/String.html)> initialPolicies)

Sets the Set of initial policy identifiers (OID strings), indicating that any one of these policies would be acceptable to the certificate user for the purposes of certification path processing. By default, any policy is acceptable (i.e. all policies), so a user that wants to allow any policy as acceptable does not need to call this method, or can call it with an empty Set (or null).

Note that the Set is copied to protect against subsequent modifications.

**Parameters:**initialPolicies - a Set of initial policy OIDs in String format (or null) **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if any of the elements in the set are not of type String**See Also:**[getInitialPolicies()](http://docs.google.com/java/security/cert/PKIXParameters.html#getInitialPolicies())

### setCertStores

public void **setCertStores**([List](http://docs.google.com/java/util/List.html)<[CertStore](http://docs.google.com/java/security/cert/CertStore.html)> stores)

Sets the list of CertStores to be used in finding certificates and CRLs. May be null, in which case no CertStores will be used. The first CertStores in the list may be preferred to those that appear later.

Note that the List is copied to protect against subsequent modifications.

**Parameters:**stores - a List of CertStores (or null) **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if any of the elements in the list are not of type java.security.cert.CertStore**See Also:**[getCertStores()](http://docs.google.com/java/security/cert/PKIXParameters.html#getCertStores())

### addCertStore

public void **addCertStore**([CertStore](http://docs.google.com/java/security/cert/CertStore.html) store)

Adds a CertStore to the end of the list of CertStores used in finding certificates and CRLs.

**Parameters:**store - the CertStore to add. If null, the store is ignored (not added to list).

### getCertStores

public [List](http://docs.google.com/java/util/List.html)<[CertStore](http://docs.google.com/java/security/cert/CertStore.html)> **getCertStores**()

Returns an immutable List of CertStores that are used to find certificates and CRLs.

**Returns:**an immutable List of CertStores (may be empty, but never null)**See Also:**[setCertStores(java.util.List)](http://docs.google.com/java/security/cert/PKIXParameters.html#setCertStores(java.util.List))

### setRevocationEnabled

public void **setRevocationEnabled**(boolean val)

Sets the RevocationEnabled flag. If this flag is true, the default revocation checking mechanism of the underlying PKIX service provider will be used. If this flag is false, the default revocation checking mechanism will be disabled (not used).

When a PKIXParameters object is created, this flag is set to true. This setting reflects the most common strategy for checking revocation, since each service provider must support revocation checking to be PKIX compliant. Sophisticated applications should set this flag to false when it is not practical to use a PKIX service provider's default revocation checking mechanism or when an alternative revocation checking mechanism is to be substituted (by also calling the [addCertPathChecker](http://docs.google.com/java/security/cert/PKIXParameters.html#addCertPathChecker(java.security.cert.PKIXCertPathChecker)) or [setCertPathCheckers](http://docs.google.com/java/security/cert/PKIXParameters.html#setCertPathCheckers(java.util.List)) methods).

**Parameters:**val - the new value of the RevocationEnabled flag

### isRevocationEnabled

public boolean **isRevocationEnabled**()

Checks the RevocationEnabled flag. If this flag is true, the default revocation checking mechanism of the underlying PKIX service provider will be used. If this flag is false, the default revocation checking mechanism will be disabled (not used). See the [setRevocationEnabled](http://docs.google.com/java/security/cert/PKIXParameters.html#setRevocationEnabled(boolean)) method for more details on setting the value of this flag.

**Returns:**the current value of the RevocationEnabled flag

### setExplicitPolicyRequired

public void **setExplicitPolicyRequired**(boolean val)

Sets the ExplicitPolicyRequired flag. If this flag is true, an acceptable policy needs to be explicitly identified in every certificate. By default, the ExplicitPolicyRequired flag is false.

**Parameters:**val - true if explicit policy is to be required, false otherwise

### isExplicitPolicyRequired

public boolean **isExplicitPolicyRequired**()

Checks if explicit policy is required. If this flag is true, an acceptable policy needs to be explicitly identified in every certificate. By default, the ExplicitPolicyRequired flag is false.

**Returns:**true if explicit policy is required, false otherwise

### setPolicyMappingInhibited

public void **setPolicyMappingInhibited**(boolean val)

Sets the PolicyMappingInhibited flag. If this flag is true, policy mapping is inhibited. By default, policy mapping is not inhibited (the flag is false).

**Parameters:**val - true if policy mapping is to be inhibited, false otherwise

### isPolicyMappingInhibited

public boolean **isPolicyMappingInhibited**()

Checks if policy mapping is inhibited. If this flag is true, policy mapping is inhibited. By default, policy mapping is not inhibited (the flag is false).

**Returns:**true if policy mapping is inhibited, false otherwise

### setAnyPolicyInhibited

public void **setAnyPolicyInhibited**(boolean val)

Sets state to determine if the any policy OID should be processed if it is included in a certificate. By default, the any policy OID is not inhibited ([isAnyPolicyInhibited()](http://docs.google.com/java/security/cert/PKIXParameters.html#isAnyPolicyInhibited()) returns false).

**Parameters:**val - true if the any policy OID is to be inhibited, false otherwise

### isAnyPolicyInhibited

public boolean **isAnyPolicyInhibited**()

Checks whether the any policy OID should be processed if it is included in a certificate.

**Returns:**true if the any policy OID is inhibited, false otherwise

### setPolicyQualifiersRejected

public void **setPolicyQualifiersRejected**(boolean qualifiersRejected)

Sets the PolicyQualifiersRejected flag. If this flag is true, certificates that include policy qualifiers in a certificate policies extension that is marked critical are rejected. If the flag is false, certificates are not rejected on this basis.

When a PKIXParameters object is created, this flag is set to true. This setting reflects the most common (and simplest) strategy for processing policy qualifiers. Applications that want to use a more sophisticated policy must set this flag to false.

Note that the PKIX certification path validation algorithm specifies that any policy qualifier in a certificate policies extension that is marked critical must be processed and validated. Otherwise the certification path must be rejected. If the policyQualifiersRejected flag is set to false, it is up to the application to validate all policy qualifiers in this manner in order to be PKIX compliant.

**Parameters:**qualifiersRejected - the new value of the PolicyQualifiersRejected flag**See Also:**[getPolicyQualifiersRejected()](http://docs.google.com/java/security/cert/PKIXParameters.html#getPolicyQualifiersRejected()), [PolicyQualifierInfo](http://docs.google.com/java/security/cert/PolicyQualifierInfo.html)

### getPolicyQualifiersRejected

public boolean **getPolicyQualifiersRejected**()

Gets the PolicyQualifiersRejected flag. If this flag is true, certificates that include policy qualifiers in a certificate policies extension that is marked critical are rejected. If the flag is false, certificates are not rejected on this basis.

When a PKIXParameters object is created, this flag is set to true. This setting reflects the most common (and simplest) strategy for processing policy qualifiers. Applications that want to use a more sophisticated policy must set this flag to false.

**Returns:**the current value of the PolicyQualifiersRejected flag**See Also:**[setPolicyQualifiersRejected(boolean)](http://docs.google.com/java/security/cert/PKIXParameters.html#setPolicyQualifiersRejected(boolean))

### getDate

public [Date](http://docs.google.com/java/util/Date.html) **getDate**()

Returns the time for which the validity of the certification path should be determined. If null, the current time is used.

Note that the Date returned is copied to protect against subsequent modifications.

**Returns:**the Date, or null if not set**See Also:**[setDate(java.util.Date)](http://docs.google.com/java/security/cert/PKIXParameters.html#setDate(java.util.Date))

### setDate

public void **setDate**([Date](http://docs.google.com/java/util/Date.html) date)

Sets the time for which the validity of the certification path should be determined. If null, the current time is used.

Note that the Date supplied here is copied to protect against subsequent modifications.

**Parameters:**date - the Date, or null for the current time**See Also:**[getDate()](http://docs.google.com/java/security/cert/PKIXParameters.html#getDate())

### setCertPathCheckers

public void **setCertPathCheckers**([List](http://docs.google.com/java/util/List.html)<[PKIXCertPathChecker](http://docs.google.com/java/security/cert/PKIXCertPathChecker.html)> checkers)

Sets a List of additional certification path checkers. If the specified List contains an object that is not a PKIXCertPathChecker, it is ignored.

Each PKIXCertPathChecker specified implements additional checks on a certificate. Typically, these are checks to process and verify private extensions contained in certificates. Each PKIXCertPathChecker should be instantiated with any initialization parameters needed to execute the check.

This method allows sophisticated applications to extend a PKIX CertPathValidator or CertPathBuilder. Each of the specified PKIXCertPathCheckers will be called, in turn, by a PKIX CertPathValidator or CertPathBuilder for each certificate processed or validated.

Regardless of whether these additional PKIXCertPathCheckers are set, a PKIX CertPathValidator or CertPathBuilder must perform all of the required PKIX checks on each certificate. The one exception to this rule is if the RevocationEnabled flag is set to false (see the [setRevocationEnabled](http://docs.google.com/java/security/cert/PKIXParameters.html#setRevocationEnabled(boolean)) method).

Note that the List supplied here is copied and each PKIXCertPathChecker in the list is cloned to protect against subsequent modifications.

**Parameters:**checkers - a List of PKIXCertPathCheckers. May be null, in which case no additional checkers will be used. **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if any of the elements in the list are not of type java.security.cert.PKIXCertPathChecker**See Also:**[getCertPathCheckers()](http://docs.google.com/java/security/cert/PKIXParameters.html#getCertPathCheckers())

### getCertPathCheckers

public [List](http://docs.google.com/java/util/List.html)<[PKIXCertPathChecker](http://docs.google.com/java/security/cert/PKIXCertPathChecker.html)> **getCertPathCheckers**()

Returns the List of certification path checkers. The returned List is immutable, and each PKIXCertPathChecker in the List is cloned to protect against subsequent modifications.

**Returns:**an immutable List of PKIXCertPathCheckers (may be empty, but not null)**See Also:**[setCertPathCheckers(java.util.List)](http://docs.google.com/java/security/cert/PKIXParameters.html#setCertPathCheckers(java.util.List))

### addCertPathChecker

public void **addCertPathChecker**([PKIXCertPathChecker](http://docs.google.com/java/security/cert/PKIXCertPathChecker.html) checker)

Adds a PKIXCertPathChecker to the list of certification path checkers. See the [setCertPathCheckers](http://docs.google.com/java/security/cert/PKIXParameters.html#setCertPathCheckers(java.util.List)) method for more details.

Note that the PKIXCertPathChecker is cloned to protect against subsequent modifications.

**Parameters:**checker - a PKIXCertPathChecker to add to the list of checks. If null, the checker is ignored (not added to list).

### getSigProvider

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

Returns the signature provider's name, or null if not set.

**Returns:**the signature provider's name (or null)**See Also:**[setSigProvider(java.lang.String)](http://docs.google.com/java/security/cert/PKIXParameters.html#setSigProvider(java.lang.String))

### setSigProvider

public void **setSigProvider**([String](http://docs.google.com/java/lang/String.html) sigProvider)

Sets the signature provider's name. The specified provider will be preferred when creating [Signature](http://docs.google.com/java/security/Signature.html) objects. If null or not set, the first provider found supporting the algorithm will be used.

**Parameters:**sigProvider - the signature provider's name (or null)**See Also:**[getSigProvider()](http://docs.google.com/java/security/cert/PKIXParameters.html#getSigProvider())

### getTargetCertConstraints

public [CertSelector](http://docs.google.com/java/security/cert/CertSelector.html) **getTargetCertConstraints**()

Returns the required constraints on the target certificate. The constraints are returned as an instance of CertSelector. If null, no constraints are defined.

Note that the CertSelector returned is cloned to protect against subsequent modifications.

**Returns:**a CertSelector specifying the constraints on the target certificate (or null)**See Also:**[setTargetCertConstraints(java.security.cert.CertSelector)](http://docs.google.com/java/security/cert/PKIXParameters.html#setTargetCertConstraints(java.security.cert.CertSelector))

### setTargetCertConstraints

public void **setTargetCertConstraints**([CertSelector](http://docs.google.com/java/security/cert/CertSelector.html) selector)

Sets the required constraints on the target certificate. The constraints are specified as an instance of CertSelector. If null, no constraints are defined.

Note that the CertSelector specified is cloned to protect against subsequent modifications.

**Parameters:**selector - a CertSelector specifying the constraints on the target certificate (or null)**See Also:**[getTargetCertConstraints()](http://docs.google.com/java/security/cert/PKIXParameters.html#getTargetCertConstraints())

### clone

public [Object](http://docs.google.com/java/lang/Object.html) **clone**()

Makes a copy of this PKIXParameters object. Changes to the copy will not affect the original and vice versa.

**Specified by:**[clone](http://docs.google.com/java/security/cert/CertPathParameters.html#clone()) in interface [CertPathParameters](http://docs.google.com/java/security/cert/CertPathParameters.html)**Overrides:**[clone](http://docs.google.com/java/lang/Object.html#clone()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a copy of this PKIXParameters object**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

### toString

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

Returns a formatted string describing the parameters.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a formatted string describing the parameters.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PKIXParameters.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/java/security/cert/PKIXCertPathValidatorResult.html)   [**NEXT CLASS**](http://docs.google.com/java/security/cert/PolicyNode.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/cert/PKIXParameters.html)    [**NO FRAMES**](http://docs.google.com/PKIXParameters.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

[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).