| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/GSSManager.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/org/ietf/jgss/GSSException.html)   [**NEXT CLASS**](http://docs.google.com/org/ietf/jgss/GSSName.html) | [**FRAMES**](http://docs.google.com/index.html?org/ietf/jgss/GSSManager.html)    [**NO FRAMES**](http://docs.google.com/GSSManager.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **org.ietf.jgss**

Class GSSManager

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **org.ietf.jgss.GSSManager**

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

This class serves as a factory for other important GSS-API classes and also provides information about the mechanisms that are supported. It can create instances of classes implementing the following three GSS-API interfaces: [GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html), [GSSCredential](http://docs.google.com/org/ietf/jgss/GSSCredential.html), and [GSSContext](http://docs.google.com/org/ietf/jgss/GSSContext.html). It also has methods to query for the list of available mechanisms and the nametypes that each mechanism supports.

An instance of the default GSSManager subclass may be obtained through the static method [getInstance](http://docs.google.com/org/ietf/jgss/GSSManager.html#getInstance()), but applications are free to instantiate other subclasses of GSSManager. The default GSSManager instance will support the Kerberos v5 GSS-API mechanism in addition to any others. This mechanism is identified by the Oid "1.2.840.113554.1.2.2" and is defined in RFC 1964.

A subclass extending the GSSManager abstract class may be implemented as a modular provider based layer that utilizes some well known service provider specification. The GSSManager API allows the application to set provider preferences on such an implementation. These methods also allow the implementation to throw a well-defined exception in case provider based configuration is not supported. Applications that expect to be portable should be aware of this and recover cleanly by catching the exception.

It is envisioned that there will be three most common ways in which providers will be used:

1. The application does not care about what provider is used (the default case).
2. The application wants a particular provider to be used preferentially, either for a particular mechanism or all the time, irrespective of mechanism.
3. The application wants to use the locally configured providers as far as possible but if support is missing for one or more mechanisms then it wants to fall back on its own provider.

The GSSManager class has two methods that enable these modes of usage: [addProviderAtFront](http://docs.google.com/org/ietf/jgss/GSSManager.html#addProviderAtFront(java.security.Provider,%20org.ietf.jgss.Oid)) and [addProviderAtEnd](http://docs.google.com/org/ietf/jgss/GSSManager.html#addProviderAtEnd(java.security.Provider,%20org.ietf.jgss.Oid)). These methods have the effect of creating an ordered list of *<provider, oid>* pairs where each pair indicates a preference of provider for a given oid.

It is important to note that there are certain interactions between the different GSS-API objects that are created by a GSSManager, where the provider that is used for a particular mechanism might need to be consistent across all objects. For instance, if a GSSCredential contains elements from a provider *p* for a mechanism *m*, it should generally be passed in to a GSSContext that will use provider *p* for the mechanism *m*. A simple rule of thumb that will maximize portability is that objects created from different GSSManager's should not be mixed, and if possible, a different GSSManager instance should be created if the application wants to invoke the addProviderAtFront method on a GSSManager that has already created an object.

Here is some sample code showing how the GSSManager might be used:

GSSManager manager = GSSManager.getInstance();  
  
 Oid krb5Mechanism = new Oid("1.2.840.113554.1.2.2");  
 Oid krb5PrincipalNameType = new Oid("1.2.840.113554.1.2.2.1");  
  
 // Identify who the client wishes to be  
 GSSName userName = manager.createName("duke", GSSName.NT\_USER\_NAME);  
  
 // Identify the name of the server. This uses a Kerberos specific  
 // name format.  
 GSSName serverName = manager.createName("nfs/foo.sun.com",   
 krb5PrincipalNameType);  
  
 // Acquire credentials for the user  
 GSSCredential userCreds = manager.createCredential(userName,  
 GSSCredential.DEFAULT\_LIFETIME,  
 krb5Mechanism,  
 GSSCredential.INITIATE\_ONLY);  
  
 // Instantiate and initialize a security context that will be  
 // established with the server  
 GSSContext context = manager.createContext(serverName,  
 krb5Mechanism,  
 userCreds,  
 GSSContext.DEFAULT\_LIFETIME);

The server side might use the following variation of this source:

// Acquire credentials for the server  
 GSSCredential serverCreds = manager.createCredential(serverName,   
 GSSCredential.DEFAULT\_LIFETIME,   
 krb5Mechanism,   
 GSSCredential.ACCEPT\_ONLY);   
   
 // Instantiate and initialize a security context that will  
 // wait for an establishment request token from the client  
 GSSContext context = manager.createContext(serverCreds);

**Since:** 1.4 **See Also:**[GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html), [GSSCredential](http://docs.google.com/org/ietf/jgss/GSSCredential.html), [GSSContext](http://docs.google.com/org/ietf/jgss/GSSContext.html)

| **Constructor Summary** | |
| --- | --- |
| [**GSSManager**](http://docs.google.com/org/ietf/jgss/GSSManager.html#GSSManager())() |

| **Method Summary** | |
| --- | --- |
| abstract  void | [**addProviderAtEnd**](http://docs.google.com/org/ietf/jgss/GSSManager.html#addProviderAtEnd(java.security.Provider,%20org.ietf.jgss.Oid))([Provider](http://docs.google.com/java/security/Provider.html) p, [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) mech)            This method is used to indicate to the GSSManager that the application would like a particular provider to be used if no other provider can be found that supports the given mechanism. |
| abstract  void | [**addProviderAtFront**](http://docs.google.com/org/ietf/jgss/GSSManager.html#addProviderAtFront(java.security.Provider,%20org.ietf.jgss.Oid))([Provider](http://docs.google.com/java/security/Provider.html) p, [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) mech)            This method is used to indicate to the GSSManager that the application would like a particular provider to be used ahead of all others when support is desired for the given mechanism. |
| abstract  [GSSContext](http://docs.google.com/org/ietf/jgss/GSSContext.html) | [**createContext**](http://docs.google.com/org/ietf/jgss/GSSManager.html#createContext(byte%5B%5D))(byte[] interProcessToken)            Factory method for creating a previously exported context. |
| abstract  [GSSContext](http://docs.google.com/org/ietf/jgss/GSSContext.html) | [**createContext**](http://docs.google.com/org/ietf/jgss/GSSManager.html#createContext(org.ietf.jgss.GSSCredential))([GSSCredential](http://docs.google.com/org/ietf/jgss/GSSCredential.html) myCred)            Factory method for creating a context on the acceptor' side. |
| abstract  [GSSContext](http://docs.google.com/org/ietf/jgss/GSSContext.html) | [**createContext**](http://docs.google.com/org/ietf/jgss/GSSManager.html#createContext(org.ietf.jgss.GSSName,%20org.ietf.jgss.Oid,%20org.ietf.jgss.GSSCredential,%20int))([GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html) peer, [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) mech, [GSSCredential](http://docs.google.com/org/ietf/jgss/GSSCredential.html) myCred, int lifetime)            Factory method for creating a context on the initiator's side. |
| abstract  [GSSCredential](http://docs.google.com/org/ietf/jgss/GSSCredential.html) | [**createCredential**](http://docs.google.com/org/ietf/jgss/GSSManager.html#createCredential(org.ietf.jgss.GSSName,%20int,%20org.ietf.jgss.Oid%5B%5D,%20int))([GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html) name, int lifetime, [Oid](http://docs.google.com/org/ietf/jgss/Oid.html)[] mechs, int usage)            Factory method for acquiring credentials over a set of mechanisms. |
| abstract  [GSSCredential](http://docs.google.com/org/ietf/jgss/GSSCredential.html) | [**createCredential**](http://docs.google.com/org/ietf/jgss/GSSManager.html#createCredential(org.ietf.jgss.GSSName,%20int,%20org.ietf.jgss.Oid,%20int))([GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html) name, int lifetime, [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) mech, int usage)            Factory method for acquiring a single mechanism credential. |
| abstract  [GSSCredential](http://docs.google.com/org/ietf/jgss/GSSCredential.html) | [**createCredential**](http://docs.google.com/org/ietf/jgss/GSSManager.html#createCredential(int))(int usage)            Factory method for acquiring default credentials. |
| abstract  [GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html) | [**createName**](http://docs.google.com/org/ietf/jgss/GSSManager.html#createName(byte%5B%5D,%20org.ietf.jgss.Oid))(byte[] name, [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) nameType)            Factory method to convert a byte array containing a name from the specified namespace to a GSSName object. |
| abstract  [GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html) | [**createName**](http://docs.google.com/org/ietf/jgss/GSSManager.html#createName(byte%5B%5D,%20org.ietf.jgss.Oid,%20org.ietf.jgss.Oid))(byte[] name, [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) nameType, [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) mech)            Factory method to convert a byte array containing a name from the specified namespace to a GSSName object and canonicalize it at the same time for a mechanism. |
| abstract  [GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html) | [**createName**](http://docs.google.com/org/ietf/jgss/GSSManager.html#createName(java.lang.String,%20org.ietf.jgss.Oid))([String](http://docs.google.com/java/lang/String.html) nameStr, [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) nameType)            Factory method to convert a string name from the specified namespace to a GSSName object. |
| abstract  [GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html) | [**createName**](http://docs.google.com/org/ietf/jgss/GSSManager.html#createName(java.lang.String,%20org.ietf.jgss.Oid,%20org.ietf.jgss.Oid))([String](http://docs.google.com/java/lang/String.html) nameStr, [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) nameType, [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) mech)            Factory method to convert a string name from the specified namespace to a GSSName object and canonicalize it at the same time for a mechanism. |
| static [GSSManager](http://docs.google.com/org/ietf/jgss/GSSManager.html) | [**getInstance**](http://docs.google.com/org/ietf/jgss/GSSManager.html#getInstance())()            Returns the default GSSManager implementation. |
| abstract  [Oid](http://docs.google.com/org/ietf/jgss/Oid.html)[] | [**getMechs**](http://docs.google.com/org/ietf/jgss/GSSManager.html#getMechs())()            Returns a list of mechanisms that are available to GSS-API callers through this GSSManager. |
| abstract  [Oid](http://docs.google.com/org/ietf/jgss/Oid.html)[] | [**getMechsForName**](http://docs.google.com/org/ietf/jgss/GSSManager.html#getMechsForName(org.ietf.jgss.Oid))([Oid](http://docs.google.com/org/ietf/jgss/Oid.html) nameType)            Returns a list of mechanisms that support the indicated name type. |
| abstract  [Oid](http://docs.google.com/org/ietf/jgss/Oid.html)[] | [**getNamesForMech**](http://docs.google.com/org/ietf/jgss/GSSManager.html#getNamesForMech(org.ietf.jgss.Oid))([Oid](http://docs.google.com/org/ietf/jgss/Oid.html) mech)            Returns then name types supported by the indicated mechanism. |

| **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** |
| --- |

### GSSManager

public **GSSManager**()

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

### getInstance

public static [GSSManager](http://docs.google.com/org/ietf/jgss/GSSManager.html) **getInstance**()

Returns the default GSSManager implementation.

**Returns:**a GSSManager implementation

### getMechs

public abstract [Oid](http://docs.google.com/org/ietf/jgss/Oid.html)[] **getMechs**()

Returns a list of mechanisms that are available to GSS-API callers through this GSSManager. The default GSSManager obtained from the [getInstance()](http://docs.google.com/org/ietf/jgss/GSSManager.html#getInstance()) method includes the Oid "1.2.840.113554.1.2.2" in its list. This Oid identifies the Kerberos v5 GSS-API mechanism that is defined in RFC 1964.

**Returns:**an array of Oid objects corresponding to the mechanisms that are available. A null value is returned when no mechanism are available (an example of this would be when mechanism are dynamically configured, and currently no mechanisms are installed).

### getNamesForMech

public abstract [Oid](http://docs.google.com/org/ietf/jgss/Oid.html)[] **getNamesForMech**([Oid](http://docs.google.com/org/ietf/jgss/Oid.html) mech)  
 throws [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html)

Returns then name types supported by the indicated mechanism.

The default GSSManager instance includes support for the Kerberos v5 mechanism. When this mechanism ("1.2.840.113554.1.2.2") is indicated, the returned list will contain at least the following nametypes: [GSSName.NT\_HOSTBASED\_SERVICE](http://docs.google.com/org/ietf/jgss/GSSName.html#NT_HOSTBASED_SERVICE), [GSSName.NT\_EXPORT\_NAME](http://docs.google.com/org/ietf/jgss/GSSName.html#NT_EXPORT_NAME), and the Kerberos v5 specific Oid "1.2.840.113554.1.2.2.1". The namespace for the Oid "1.2.840.113554.1.2.2.1" is defined in RFC 1964.

**Parameters:**mech - the Oid of the mechanism to query **Returns:**an array of Oid objects corresponding to the name types that the mechanism supports. **Throws:** [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html) - containing the following major error codes: [GSSException.BAD\_MECH](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_MECH) [GSSException.FAILURE](http://docs.google.com/org/ietf/jgss/GSSException.html#FAILURE)**See Also:**[getMechsForName(Oid)](http://docs.google.com/org/ietf/jgss/GSSManager.html#getMechsForName(org.ietf.jgss.Oid))

### getMechsForName

public abstract [Oid](http://docs.google.com/org/ietf/jgss/Oid.html)[] **getMechsForName**([Oid](http://docs.google.com/org/ietf/jgss/Oid.html) nameType)

Returns a list of mechanisms that support the indicated name type.

The Kerberos v5 mechanism ("1.2.840.113554.1.2.2") will always be returned in this list when the indicated nametype is one of [GSSName.NT\_HOSTBASED\_SERVICE](http://docs.google.com/org/ietf/jgss/GSSName.html#NT_HOSTBASED_SERVICE), [GSSName.NT\_EXPORT\_NAME](http://docs.google.com/org/ietf/jgss/GSSName.html#NT_EXPORT_NAME), or "1.2.840.113554.1.2.2.1".

**Parameters:**nameType - the Oid of the name type to look for **Returns:**an array of Oid objects corresponding to the mechanisms that support the specified name type. null is returned when no mechanisms are found to support the specified name type.**See Also:**[getNamesForMech(Oid)](http://docs.google.com/org/ietf/jgss/GSSManager.html#getNamesForMech(org.ietf.jgss.Oid))

### createName

public abstract [GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html) **createName**([String](http://docs.google.com/java/lang/String.html) nameStr,  
 [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) nameType)  
 throws [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html)

Factory method to convert a string name from the specified namespace to a GSSName object. In general, the GSSName object created will contain multiple representations of the name, one for each mechanism that is supported; two examples that are exceptions to this are when the namespace type parameter indicates NT\_EXPORT\_NAME or when the GSS-API implementation is not multi-mechanism. It is not recommended to use this method with a NT\_EXPORT\_NAME type because representing a previously exported name consisting of abitrary bytes as a String might cause problems with character encoding schemes. In such cases it is recommended that the bytes be passed in directly to the overloaded form of this method [createName](http://docs.google.com/org/ietf/jgss/GSSManager.html#createName(byte%5B%5D,%20org.ietf.jgss.Oid)).

**Parameters:**nameStr - the string representing a printable form of the name to create.nameType - the Oid specifying the namespace of the printable name supplied. null can be used to specify that a mechanism specific default printable syntax should be assumed by each mechanism that examines nameStr. It is not advisable to use the nametype NT\_EXPORT\_NAME with this method. **Returns:**a GSSName representing the indicated principal **Throws:** [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html) - containing the following major error codes: [GSSException.BAD\_NAMETYPE](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_NAMETYPE), [GSSException.BAD\_NAME](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_NAME), [GSSException.BAD\_MECH](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_MECH), [GSSException.FAILURE](http://docs.google.com/org/ietf/jgss/GSSException.html#FAILURE)**See Also:**[GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html), [GSSName.NT\_EXPORT\_NAME](http://docs.google.com/org/ietf/jgss/GSSName.html#NT_EXPORT_NAME)

### createName

public abstract [GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html) **createName**(byte[] name,  
 [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) nameType)  
 throws [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html)

Factory method to convert a byte array containing a name from the specified namespace to a GSSName object. In general, the GSSName object created will contain multiple representations of the name, one for each mechanism that is supported; two examples that are exceptions to this are when the namespace type parameter indicates NT\_EXPORT\_NAME or when the GSS-API implementation is not multi-mechanism. The bytes that are passed in are interpreted by each underlying mechanism according to some encoding scheme of its choice for the given nametype.

**Parameters:**name - the byte array containing the name to createnameType - the Oid specifying the namespace of the name supplied in the byte array. null can be used to specify that a mechanism specific default syntax should be assumed by each mechanism that examines the byte array. **Returns:**a GSSName representing the indicated principal **Throws:** [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html) - containing the following major error codes: [GSSException.BAD\_NAMETYPE](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_NAMETYPE), [GSSException.BAD\_NAME](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_NAME), [GSSException.BAD\_MECH](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_MECH), [GSSException.FAILURE](http://docs.google.com/org/ietf/jgss/GSSException.html#FAILURE)**See Also:**[GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html), [GSSName.NT\_EXPORT\_NAME](http://docs.google.com/org/ietf/jgss/GSSName.html#NT_EXPORT_NAME)

### createName

public abstract [GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html) **createName**([String](http://docs.google.com/java/lang/String.html) nameStr,  
 [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) nameType,  
 [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) mech)  
 throws [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html)

Factory method to convert a string name from the specified namespace to a GSSName object and canonicalize it at the same time for a mechanism. In other words, this method is a utility that does the equivalent of two steps: the [createName](http://docs.google.com/org/ietf/jgss/GSSManager.html#createName(java.lang.String,%20org.ietf.jgss.Oid)) and then also the [GSSName.canonicalize](http://docs.google.com/org/ietf/jgss/GSSName.html#canonicalize(org.ietf.jgss.Oid)).

**Parameters:**nameStr - the string representing a printable form of the name to create.nameType - the Oid specifying the namespace of the printable name supplied. null can be used to specify that a mechanism specific default printable syntax should be assumed by each mechanism that examines nameStr. It is not advisable to use the nametype NT\_EXPORT\_NAME with this method.mech - Oid specifying the mechanism for which the name should be canonicalized **Returns:**a GSSName representing the indicated principal **Throws:** [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html) - containing the following major error codes: [GSSException.BAD\_NAMETYPE](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_NAMETYPE), [GSSException.BAD\_NAME](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_NAME), [GSSException.BAD\_MECH](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_MECH), [GSSException.FAILURE](http://docs.google.com/org/ietf/jgss/GSSException.html#FAILURE)**See Also:**[GSSName.canonicalize(Oid)](http://docs.google.com/org/ietf/jgss/GSSName.html#canonicalize(org.ietf.jgss.Oid)), [GSSName.NT\_EXPORT\_NAME](http://docs.google.com/org/ietf/jgss/GSSName.html#NT_EXPORT_NAME)

### createName

public abstract [GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html) **createName**(byte[] name,  
 [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) nameType,  
 [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) mech)  
 throws [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html)

Factory method to convert a byte array containing a name from the specified namespace to a GSSName object and canonicalize it at the same time for a mechanism. In other words, this method is a utility that does the equivalent of two steps: the [createName](http://docs.google.com/org/ietf/jgss/GSSManager.html#createName(byte%5B%5D,%20org.ietf.jgss.Oid)) and then also [GSSName.canonicalize](http://docs.google.com/org/ietf/jgss/GSSName.html#canonicalize(org.ietf.jgss.Oid)).

**Parameters:**name - the byte array containing the name to createnameType - the Oid specifying the namespace of the name supplied in the byte array. null can be used to specify that a mechanism specific default syntax should be assumed by each mechanism that examines the byte array.mech - Oid specifying the mechanism for which the name should be canonicalized **Returns:**a GSSName representing the indicated principal **Throws:** [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html) - containing the following major error codes: [GSSException.BAD\_NAMETYPE](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_NAMETYPE), [GSSException.BAD\_NAME](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_NAME), [GSSException.BAD\_MECH](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_MECH), [GSSException.FAILURE](http://docs.google.com/org/ietf/jgss/GSSException.html#FAILURE)**See Also:**[GSSName.canonicalize(Oid)](http://docs.google.com/org/ietf/jgss/GSSName.html#canonicalize(org.ietf.jgss.Oid)), [GSSName.NT\_EXPORT\_NAME](http://docs.google.com/org/ietf/jgss/GSSName.html#NT_EXPORT_NAME)

### createCredential

public abstract [GSSCredential](http://docs.google.com/org/ietf/jgss/GSSCredential.html) **createCredential**(int usage)  
 throws [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html)

Factory method for acquiring default credentials. This will cause the GSS-API to use system specific defaults for the set of mechanisms, name, and lifetime.

GSS-API mechanism providers must impose a local access-control policy on callers to prevent unauthorized callers from acquiring credentials to which they are not entitled. The kinds of permissions needed by different mechanism providers will be documented on a per-mechanism basis. A failed permission check might cause a [SecurityException](http://docs.google.com/java/lang/SecurityException.html) to be thrown from this method.

**Parameters:**usage - The intended usage for this credential object. The value of this parameter must be one of: [GSSCredential.INITIATE\_AND\_ACCEPT](http://docs.google.com/org/ietf/jgss/GSSCredential.html#INITIATE_AND_ACCEPT), [GSSCredential.ACCEPT\_ONLY](http://docs.google.com/org/ietf/jgss/GSSCredential.html#ACCEPT_ONLY), and [GSSCredential.INITIATE\_ONLY](http://docs.google.com/org/ietf/jgss/GSSCredential.html#INITIATE_ONLY). **Returns:**a GSSCredential of the requested type. **Throws:** [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html) - containing the following major error codes: [GSSException.BAD\_MECH](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_MECH), [GSSException.BAD\_NAMETYPE](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_NAMETYPE), [GSSException.BAD\_NAME](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_NAME), [GSSException.CREDENTIALS\_EXPIRED](http://docs.google.com/org/ietf/jgss/GSSException.html#CREDENTIALS_EXPIRED), [GSSException.NO\_CRED](http://docs.google.com/org/ietf/jgss/GSSException.html#NO_CRED), [GSSException.FAILURE](http://docs.google.com/org/ietf/jgss/GSSException.html#FAILURE)**See Also:**[GSSCredential](http://docs.google.com/org/ietf/jgss/GSSCredential.html)

### createCredential

public abstract [GSSCredential](http://docs.google.com/org/ietf/jgss/GSSCredential.html) **createCredential**([GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html) name,  
 int lifetime,  
 [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) mech,  
 int usage)  
 throws [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html)

Factory method for acquiring a single mechanism credential.

GSS-API mechanism providers must impose a local access-control policy on callers to prevent unauthorized callers from acquiring credentials to which they are not entitled. The kinds of permissions needed by different mechanism providers will be documented on a per-mechanism basis. A failed permission check might cause a [SecurityException](http://docs.google.com/java/lang/SecurityException.html) to be thrown from this method.

Non-default values for lifetime cannot always be honored by the underlying mechanisms, thus applications should be prepared to call [getRemainingLifetime](http://docs.google.com/org/ietf/jgss/GSSCredential.html#getRemainingLifetime()) on the returned credential.

**Parameters:**name - the name of the principal for whom this credential is to be acquired. Use null to specify the default principal.lifetime - The number of seconds that credentials should remain valid. Use [GSSCredential.INDEFINITE\_LIFETIME](http://docs.google.com/org/ietf/jgss/GSSCredential.html#INDEFINITE_LIFETIME) to request that the credentials have the maximum permitted lifetime. Use [GSSCredential.DEFAULT\_LIFETIME](http://docs.google.com/org/ietf/jgss/GSSCredential.html#DEFAULT_LIFETIME) to request default credential lifetime.mech - the Oid of the desired mechanism. Use (Oid) null to request the default mechanism.usage - The intended usage for this credential object. The value of this parameter must be one of: [GSSCredential.INITIATE\_AND\_ACCEPT](http://docs.google.com/org/ietf/jgss/GSSCredential.html#INITIATE_AND_ACCEPT), [GSSCredential.ACCEPT\_ONLY](http://docs.google.com/org/ietf/jgss/GSSCredential.html#ACCEPT_ONLY), and [GSSCredential.INITIATE\_ONLY](http://docs.google.com/org/ietf/jgss/GSSCredential.html#INITIATE_ONLY). **Returns:**a GSSCredential of the requested type. **Throws:** [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html) - containing the following major error codes: [GSSException.BAD\_MECH](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_MECH), [GSSException.BAD\_NAMETYPE](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_NAMETYPE), [GSSException.BAD\_NAME](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_NAME), [GSSException.CREDENTIALS\_EXPIRED](http://docs.google.com/org/ietf/jgss/GSSException.html#CREDENTIALS_EXPIRED), [GSSException.NO\_CRED](http://docs.google.com/org/ietf/jgss/GSSException.html#NO_CRED), [GSSException.FAILURE](http://docs.google.com/org/ietf/jgss/GSSException.html#FAILURE)**See Also:**[GSSCredential](http://docs.google.com/org/ietf/jgss/GSSCredential.html)

### createCredential

public abstract [GSSCredential](http://docs.google.com/org/ietf/jgss/GSSCredential.html) **createCredential**([GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html) name,  
 int lifetime,  
 [Oid](http://docs.google.com/org/ietf/jgss/Oid.html)[] mechs,  
 int usage)  
 throws [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html)

Factory method for acquiring credentials over a set of mechanisms. This method attempts to acquire credentials for each of the mechanisms specified in the array called mechs. To determine the list of mechanisms for which the acquisition of credentials succeeded, the caller should use the [GSSCredential.getMechs](http://docs.google.com/org/ietf/jgss/GSSCredential.html#getMechs()) method.

GSS-API mechanism providers must impose a local access-control policy on callers to prevent unauthorized callers from acquiring credentials to which they are not entitled. The kinds of permissions needed by different mechanism providers will be documented on a per-mechanism basis. A failed permission check might cause a [SecurityException](http://docs.google.com/java/lang/SecurityException.html) to be thrown from this method.

Non-default values for lifetime cannot always be honored by the underlying mechanisms, thus applications should be prepared to call [getRemainingLifetime](http://docs.google.com/org/ietf/jgss/GSSCredential.html#getRemainingLifetime()) on the returned credential.

**Parameters:**name - the name of the principal for whom this credential is to be acquired. Use null to specify the default principal.lifetime - The number of seconds that credentials should remain valid. Use [GSSCredential.INDEFINITE\_LIFETIME](http://docs.google.com/org/ietf/jgss/GSSCredential.html#INDEFINITE_LIFETIME) to request that the credentials have the maximum permitted lifetime. Use [GSSCredential.DEFAULT\_LIFETIME](http://docs.google.com/org/ietf/jgss/GSSCredential.html#DEFAULT_LIFETIME) to request default credential lifetime.mechs - an array of Oid's indicating the mechanisms over which the credential is to be acquired. Use (Oid[]) null for requesting a system specific default set of mechanisms.usage - The intended usage for this credential object. The value of this parameter must be one of: [GSSCredential.INITIATE\_AND\_ACCEPT](http://docs.google.com/org/ietf/jgss/GSSCredential.html#INITIATE_AND_ACCEPT), [GSSCredential.ACCEPT\_ONLY](http://docs.google.com/org/ietf/jgss/GSSCredential.html#ACCEPT_ONLY), and [GSSCredential.INITIATE\_ONLY](http://docs.google.com/org/ietf/jgss/GSSCredential.html#INITIATE_ONLY). **Returns:**a GSSCredential of the requested type. **Throws:** [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html) - containing the following major error codes: [GSSException.BAD\_MECH](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_MECH), [GSSException.BAD\_NAMETYPE](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_NAMETYPE), [GSSException.BAD\_NAME](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_NAME), [GSSException.CREDENTIALS\_EXPIRED](http://docs.google.com/org/ietf/jgss/GSSException.html#CREDENTIALS_EXPIRED), [GSSException.NO\_CRED](http://docs.google.com/org/ietf/jgss/GSSException.html#NO_CRED), [GSSException.FAILURE](http://docs.google.com/org/ietf/jgss/GSSException.html#FAILURE)**See Also:**[GSSCredential](http://docs.google.com/org/ietf/jgss/GSSCredential.html)

### createContext

public abstract [GSSContext](http://docs.google.com/org/ietf/jgss/GSSContext.html) **createContext**([GSSName](http://docs.google.com/org/ietf/jgss/GSSName.html) peer,  
 [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) mech,  
 [GSSCredential](http://docs.google.com/org/ietf/jgss/GSSCredential.html) myCred,  
 int lifetime)  
 throws [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html)

Factory method for creating a context on the initiator's side. Some mechanism providers might require that the caller be granted permission to initiate a security context. A failed permission check might cause a [SecurityException](http://docs.google.com/java/lang/SecurityException.html) to be thrown from this method.

Non-default values for lifetime cannot always be honored by the underlying mechanism, thus applications should be prepared to call [getLifetime](http://docs.google.com/org/ietf/jgss/GSSContext.html#getLifetime()) on the returned context.

**Parameters:**peer - the name of the target peer.mech - the Oid of the desired mechanism. Use null to request the default mechanism.myCred - the credentials of the initiator. Use null to act as the default initiator principal.lifetime - the lifetime, in seconds, requested for the context. Use [GSSContext.INDEFINITE\_LIFETIME](http://docs.google.com/org/ietf/jgss/GSSContext.html#INDEFINITE_LIFETIME) to request that the context have the maximum permitted lifetime. Use [GSSContext.DEFAULT\_LIFETIME](http://docs.google.com/org/ietf/jgss/GSSContext.html#DEFAULT_LIFETIME) to request a default lifetime for the context. **Returns:**an unestablished GSSContext **Throws:** [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html) - containing the following major error codes: [GSSException.NO\_CRED](http://docs.google.com/org/ietf/jgss/GSSException.html#NO_CRED) [GSSException.CREDENTIALS\_EXPIRED](http://docs.google.com/org/ietf/jgss/GSSException.html#CREDENTIALS_EXPIRED) [GSSException.BAD\_NAMETYPE](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_NAMETYPE) [GSSException.BAD\_MECH](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_MECH) [GSSException.FAILURE](http://docs.google.com/org/ietf/jgss/GSSException.html#FAILURE)**See Also:**[GSSContext](http://docs.google.com/org/ietf/jgss/GSSContext.html)

### createContext

public abstract [GSSContext](http://docs.google.com/org/ietf/jgss/GSSContext.html) **createContext**([GSSCredential](http://docs.google.com/org/ietf/jgss/GSSCredential.html) myCred)  
 throws [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html)

Factory method for creating a context on the acceptor' side. The context's properties will be determined from the input token supplied to the accept method. Some mechanism providers might require that the caller be granted permission to accept a security context. A failed permission check might cause a [SecurityException](http://docs.google.com/java/lang/SecurityException.html) to be thrown from this method.

**Parameters:**myCred - the credentials for the acceptor. Use null to act as a default acceptor principal. **Returns:**an unestablished GSSContext **Throws:** [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html) - containing the following major error codes: [GSSException.NO\_CRED](http://docs.google.com/org/ietf/jgss/GSSException.html#NO_CRED) [GSSException.CREDENTIALS\_EXPIRED](http://docs.google.com/org/ietf/jgss/GSSException.html#CREDENTIALS_EXPIRED) [GSSException.BAD\_MECH](http://docs.google.com/org/ietf/jgss/GSSException.html#BAD_MECH) [GSSException.FAILURE](http://docs.google.com/org/ietf/jgss/GSSException.html#FAILURE)**See Also:**[GSSContext](http://docs.google.com/org/ietf/jgss/GSSContext.html)

### createContext

public abstract [GSSContext](http://docs.google.com/org/ietf/jgss/GSSContext.html) **createContext**(byte[] interProcessToken)  
 throws [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html)

Factory method for creating a previously exported context. The context properties will be determined from the input token and cannot be modified through the set methods.

Implementations are not required to support the inter-process transfer of security contexts. Before exporting a context, calling the [GSSContext.isTransferable](http://docs.google.com/org/ietf/jgss/GSSContext.html#isTransferable()) will indicate if the context is transferable. Calling this method in an implementation that does not support it will result in a GSSException with the error code [GSSException.UNAVAILABLE](http://docs.google.com/org/ietf/jgss/GSSException.html#UNAVAILABLE). Some mechanism providers might require that the caller be granted permission to initiate or accept a security context. A failed permission check might cause a [SecurityException](http://docs.google.com/java/lang/SecurityException.html) to be thrown from this method.

**Parameters:**interProcessToken - the token previously emitted from the export method. **Returns:**the previously established GSSContext **Throws:** [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html) - containing the following major error codes: [GSSException.NO\_CONTEXT](http://docs.google.com/org/ietf/jgss/GSSException.html#NO_CONTEXT), [GSSException.DEFECTIVE\_TOKEN](http://docs.google.com/org/ietf/jgss/GSSException.html#DEFECTIVE_TOKEN), [GSSException.UNAVAILABLE](http://docs.google.com/org/ietf/jgss/GSSException.html#UNAVAILABLE), [GSSException.UNAUTHORIZED](http://docs.google.com/org/ietf/jgss/GSSException.html#UNAUTHORIZED), [GSSException.FAILURE](http://docs.google.com/org/ietf/jgss/GSSException.html#FAILURE)**See Also:**[GSSContext](http://docs.google.com/org/ietf/jgss/GSSContext.html)

### addProviderAtFront

public abstract void **addProviderAtFront**([Provider](http://docs.google.com/java/security/Provider.html) p,  
 [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) mech)  
 throws [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html)

This method is used to indicate to the GSSManager that the application would like a particular provider to be used ahead of all others when support is desired for the given mechanism. When a value of null is used instead of an Oid for the mechanism, the GSSManager must use the indicated provider ahead of all others no matter what the mechanism is. Only when the indicated provider does not support the needed mechanism should the GSSManager move on to a different provider.

Calling this method repeatedly preserves the older settings but lowers them in preference thus forming an ordered list of provider and Oid pairs that grows at the top.

Calling addProviderAtFront with a null Oid will remove all previous preferences that were set for this provider in the GSSManager instance. Calling addProviderAtFront with a non-null Oid will remove any previous preference that was set using this mechanism and this provider together.

If the GSSManager implementation does not support an SPI with a pluggable provider architecture it should throw a GSSException with the status code GSSException.UNAVAILABLE to indicate that the operation is unavailable.

Suppose an application desired that the provider A always be checked first when any mechanism is needed, it would call:

GSSManager mgr = GSSManager.getInstance();  
 // mgr may at this point have its own pre-configured list  
 // of provider preferences. The following will prepend to  
 // any such list:  
  
 mgr.addProviderAtFront(A, null);

Now if it also desired that the mechanism of Oid m1 always be obtained from the provider B before the previously set A was checked, it would call:

mgr.addProviderAtFront(B, m1);

The GSSManager would then first check with B if m1 was needed. In case B did not provide support for m1, the GSSManager would continue on to check with A. If any mechanism m2 is needed where m2 is different from m1 then the GSSManager would skip B and check with A directly.

Suppose at a later time the following call is made to the same GSSManager instance:

mgr.addProviderAtFront(B, null)

then the previous setting with the pair (B, m1) is subsumed by this and should be removed. Effectively the list of preferences now becomes {(B, null), (A, null), ... //followed by the pre-configured list.

Please note, however, that the following call:

mgr.addProviderAtFront(A, m3)

does not subsume the previous setting of (A, null) and the list will effectively become {(A, m3), (B, null), (A, null), ...}

**Parameters:**p - the provider instance that should be used whenever support is needed for mech.mech - the mechanism for which the provider is being set **Throws:** [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html) - containing the following major error codes: [GSSException.UNAVAILABLE](http://docs.google.com/org/ietf/jgss/GSSException.html#UNAVAILABLE), [GSSException.FAILURE](http://docs.google.com/org/ietf/jgss/GSSException.html#FAILURE)

### addProviderAtEnd

public abstract void **addProviderAtEnd**([Provider](http://docs.google.com/java/security/Provider.html) p,  
 [Oid](http://docs.google.com/org/ietf/jgss/Oid.html) mech)  
 throws [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html)

This method is used to indicate to the GSSManager that the application would like a particular provider to be used if no other provider can be found that supports the given mechanism. When a value of null is used instead of an Oid for the mechanism, the GSSManager must use the indicated provider for any mechanism.

Calling this method repeatedly preserves the older settings but raises them above newer ones in preference thus forming an ordered list of providers and Oid pairs that grows at the bottom. Thus the older provider settings will be utilized first before this one is.

If there are any previously existing preferences that conflict with the preference being set here, then the GSSManager should ignore this request.

If the GSSManager implementation does not support an SPI with a pluggable provider architecture it should throw a GSSException with the status code GSSException.UNAVAILABLE to indicate that the operation is unavailable.

Suppose an application desired that when a mechanism of Oid m1 is needed the system default providers always be checked first, and only when they do not support m1 should a provider A be checked. It would then make the call:

GSSManager mgr = GSSManager.getInstance();  
 mgr.addProviderAtEnd(A, m1);

Now, if it also desired that for all mechanisms the provider B be checked after all configured providers have been checked, it would then call:

mgr.addProviderAtEnd(B, null);

Effectively the list of preferences now becomes {..., (A, m1), (B, null)}.

Suppose at a later time the following call is made to the same GSSManager instance:

mgr.addProviderAtEnd(B, m2)

then the previous setting with the pair (B, null) subsumes this and therefore this request should be ignored. The same would happen if a request is made for the already existing pairs of (A, m1) or (B, null).

Please note, however, that the following call:

mgr.addProviderAtEnd(A, null)

is not subsumed by the previous setting of (A, m1) and the list will effectively become {..., (A, m1), (B, null), (A, null)}

**Parameters:**p - the provider instance that should be used whenever support is needed for mech.mech - the mechanism for which the provider is being set **Throws:** [GSSException](http://docs.google.com/org/ietf/jgss/GSSException.html) - containing the following major error codes: [GSSException.UNAVAILABLE](http://docs.google.com/org/ietf/jgss/GSSException.html#UNAVAILABLE), [GSSException.FAILURE](http://docs.google.com/org/ietf/jgss/GSSException.html#FAILURE)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/GSSManager.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/org/ietf/jgss/GSSException.html)   [**NEXT CLASS**](http://docs.google.com/org/ietf/jgss/GSSName.html) | [**FRAMES**](http://docs.google.com/index.html?org/ietf/jgss/GSSManager.html)    [**NO FRAMES**](http://docs.google.com/GSSManager.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).