| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ServiceRegistry.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/imageio/spi/RegisterableService.html)   [**NEXT CLASS**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.Filter.html) | [**FRAMES**](http://docs.google.com/index.html?javax/imageio/spi/ServiceRegistry.html)    [**NO FRAMES**](http://docs.google.com/ServiceRegistry.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#2s8eyo1) |

## **javax.imageio.spi**

Class ServiceRegistry

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.imageio.spi.ServiceRegistry**

**Direct Known Subclasses:** [IIORegistry](http://docs.google.com/javax/imageio/spi/IIORegistry.html)

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

A registry for service provider instances.

A *service* is a well-known set of interfaces and (usually abstract) classes. A *service provider* is a specific implementation of a service. The classes in a provider typically implement the interface or subclass the class defined by the service itself.

Service providers are stored in one or more *categories*, each of which is defined by a class of interface (described by a Class object) that all of its members must implement. The set of categories may be changed dynamically.

Only a single instance of a given leaf class (that is, the actual class returned by getClass(), as opposed to any inherited classes or interfaces) may be registered. That is, suppose that the com.mycompany.mypkg.GreenServiceProvider class implements the com.mycompany.mypkg.MyService interface. If a GreenServiceProvider instance is registered, it will be stored in the category defined by the MyService class. If a new instance of GreenServiceProvider is registered, it will replace the previous instance. In practice, service provider objects are usually singletons so this behavior is appropriate.

To declare a service provider, a services subdirectory is placed within the META-INF directory that is present in every JAR file. This directory contains a file for each service provider interface that has one or more implementation classes present in the JAR file. For example, if the JAR file contained a class named com.mycompany.mypkg.MyServiceImpl which implements the javax.someapi.SomeService interface, the JAR file would contain a file named:

META-INF/services/javax.someapi.SomeService

containing the line:

com.mycompany.mypkg.MyService

The service provider classes should be to be lightweight and quick to load. Implementations of these interfaces should avoid complex dependencies on other classes and on native code. The usual pattern for more complex services is to register a lightweight proxy for the heavyweight service.

An application may customize the contents of a registry as it sees fit, so long as it has the appropriate runtime permission.

For more details on declaring service providers, and the JAR format in general, see the  [JAR File Specification](http://docs.google.com/technotes/guides/jar/jar.html).

**See Also:**[RegisterableService](http://docs.google.com/javax/imageio/spi/RegisterableService.html)

| **Nested Class Summary** | |
| --- | --- |
| static interface | [**ServiceRegistry.Filter**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.Filter.html)            A simple filter interface used by ServiceRegistry.getServiceProviders to select providers matching an arbitrary criterion. |

| **Constructor Summary** | |
| --- | --- |
| [**ServiceRegistry**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#ServiceRegistry(java.util.Iterator))([Iterator](http://docs.google.com/java/util/Iterator.html)<[Class](http://docs.google.com/java/lang/Class.html)<?>> categories)            Constructs a ServiceRegistry instance with a set of categories taken from the categories argument. |

| **Method Summary** | |
| --- | --- |
| boolean | [**contains**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#contains(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) provider)            Returns true if provider is currently registered. |
| void | [**deregisterAll**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#deregisterAll())()            Deregisters all currently registered service providers from all categories. |
| void | [**deregisterAll**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#deregisterAll(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<?> category)            Deregisters all service provider object currently registered under the given category. |
| void | [**deregisterServiceProvider**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#deregisterServiceProvider(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) provider)            Removes a service provider object from all categories that contain it. |
| | <T> boolean | | --- | | [**deregisterServiceProvider**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#deregisterServiceProvider(T,%20java.lang.Class))(T provider, [Class](http://docs.google.com/java/lang/Class.html)<T> category)            Removes a service provider object from the given category. |
| void | [**finalize**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#finalize())()            Finalizes this object prior to garbage collection. |
| [Iterator](http://docs.google.com/java/util/Iterator.html)<[Class](http://docs.google.com/java/lang/Class.html)<?>> | [**getCategories**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#getCategories())()            Returns an Iterator of Class objects indicating the current set of categories. |
| | <T> T | | --- | | [**getServiceProviderByClass**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#getServiceProviderByClass(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<T> providerClass)            Returns the currently registered service provider object that is of the given class type. |
| | <T> [Iterator](http://docs.google.com/java/util/Iterator.html)<T> | | --- | | [**getServiceProviders**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#getServiceProviders(java.lang.Class,%20boolean))([Class](http://docs.google.com/java/lang/Class.html)<T> category, boolean useOrdering)            Returns an Iterator containing all registered service providers in the given category. |
| | <T> [Iterator](http://docs.google.com/java/util/Iterator.html)<T> | | --- | | [**getServiceProviders**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#getServiceProviders(java.lang.Class,%20javax.imageio.spi.ServiceRegistry.Filter,%20boolean))([Class](http://docs.google.com/java/lang/Class.html)<T> category, [ServiceRegistry.Filter](http://docs.google.com/javax/imageio/spi/ServiceRegistry.Filter.html) filter, boolean useOrdering)            Returns an Iterator containing service provider objects within a given category that satisfy a criterion imposed by the supplied ServiceRegistry.Filter object's filter method. |
| static   | <T> [Iterator](http://docs.google.com/java/util/Iterator.html)<T> | | --- | | [**lookupProviders**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#lookupProviders(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<T> providerClass)            Locates and incrementally instantiates the available providers of a given service using the context class loader. |
| static   | <T> [Iterator](http://docs.google.com/java/util/Iterator.html)<T> | | --- | | [**lookupProviders**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#lookupProviders(java.lang.Class,%20java.lang.ClassLoader))([Class](http://docs.google.com/java/lang/Class.html)<T> providerClass, [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) loader)            Searches for implementations of a particular service class using the given class loader. |
| void | [**registerServiceProvider**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#registerServiceProvider(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) provider)            Adds a service provider object to the registry. |
| | <T> boolean | | --- | | [**registerServiceProvider**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#registerServiceProvider(T,%20java.lang.Class))(T provider, [Class](http://docs.google.com/java/lang/Class.html)<T> category)            Adds a service provider object to the registry. |
| void | [**registerServiceProviders**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#registerServiceProviders(java.util.Iterator))([Iterator](http://docs.google.com/java/util/Iterator.html)<?> providers)            Adds a set of service provider objects, taken from an Iterator to the registry. |
| | <T> boolean | | --- | | [**setOrdering**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#setOrdering(java.lang.Class,%20T,%20T))([Class](http://docs.google.com/java/lang/Class.html)<T> category, T firstProvider, T secondProvider)            Sets a pairwise ordering between two service provider objects within a given category. |
| | <T> boolean | | --- | | [**unsetOrdering**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.html#unsetOrdering(java.lang.Class,%20T,%20T))([Class](http://docs.google.com/java/lang/Class.html)<T> category, T firstProvider, T secondProvider)            Sets a pairwise ordering between two service provider objects within a given category. |

| **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)), [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** |
| --- |

### ServiceRegistry

public **ServiceRegistry**([Iterator](http://docs.google.com/java/util/Iterator.html)<[Class](http://docs.google.com/java/lang/Class.html)<?>> categories)

Constructs a ServiceRegistry instance with a set of categories taken from the categories argument.

**Parameters:**categories - an Iterator containing Class objects to be used to define categories. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if categories is null.

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

### lookupProviders

public static <T> [Iterator](http://docs.google.com/java/util/Iterator.html)<T> **lookupProviders**([Class](http://docs.google.com/java/lang/Class.html)<T> providerClass,  
 [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) loader)

Searches for implementations of a particular service class using the given class loader.

This method transforms the name of the given service class into a provider-configuration filename as described in the class comment and then uses the getResources method of the given class loader to find all available files with that name. These files are then read and parsed to produce a list of provider-class names. The iterator that is returned uses the given class loader to look up and then instantiate each element of the list.

Because it is possible for extensions to be installed into a running Java virtual machine, this method may return different results each time it is invoked.

**Parameters:**providerClass - a Classobject indicating the class or interface of the service providers being detected.loader - the class loader to be used to load provider-configuration files and instantiate provider classes, or null if the system class loader (or, failing that the bootstrap class loader) is to be used. **Returns:**An Iterator that yields provider objects for the given service, in some arbitrary order. The iterator will throw an Error if a provider-configuration file violates the specified format or if a provider class cannot be found and instantiated. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if providerClass is null.

### lookupProviders

public static <T> [Iterator](http://docs.google.com/java/util/Iterator.html)<T> **lookupProviders**([Class](http://docs.google.com/java/lang/Class.html)<T> providerClass)

Locates and incrementally instantiates the available providers of a given service using the context class loader. This convenience method is equivalent to:

ClassLoader cl = Thread.currentThread().getContextClassLoader();  
 return Service.providers(service, cl);

**Parameters:**providerClass - a Classobject indicating the class or interface of the service providers being detected. **Returns:**An Iterator that yields provider objects for the given service, in some arbitrary order. The iterator will throw an Error if a provider-configuration file violates the specified format or if a provider class cannot be found and instantiated. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if providerClass is null.

### getCategories

public [Iterator](http://docs.google.com/java/util/Iterator.html)<[Class](http://docs.google.com/java/lang/Class.html)<?>> **getCategories**()

Returns an Iterator of Class objects indicating the current set of categories. The iterator will be empty if no categories exist.

**Returns:**an Iterator containing Classobjects.

### registerServiceProvider

public <T> boolean **registerServiceProvider**(T provider,  
 [Class](http://docs.google.com/java/lang/Class.html)<T> category)

Adds a service provider object to the registry. The provider is associated with the given category.

If provider implements the RegisterableService interface, its onRegistration method will be called. Its onDeregistration method will be called each time it is deregistered from a category, for example if a category is removed or the registry is garbage collected.

**Parameters:**provider - the service provide object to be registered.category - the category under which to register the provider. **Returns:**true if no provider of the same class was previously registered in the same category category. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if provider is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if there is no category corresponding to category. [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if provider does not implement the Class defined by category.

### registerServiceProvider

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

Adds a service provider object to the registry. The provider is associated within each category present in the registry whose Class it implements.

If provider implements the RegisterableService interface, its onRegistration method will be called once for each category it is registered under. Its onDeregistration method will be called each time it is deregistered from a category or when the registry is finalized.

**Parameters:**provider - the service provider object to be registered. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if provider is null.

### registerServiceProviders

public void **registerServiceProviders**([Iterator](http://docs.google.com/java/util/Iterator.html)<?> providers)

Adds a set of service provider objects, taken from an Iterator to the registry. Each provider is associated within each category present in the registry whose Class it implements.

For each entry of providers that implements the RegisterableService interface, its onRegistration method will be called once for each category it is registered under. Its onDeregistration method will be called each time it is deregistered from a category or when the registry is finalized.

**Parameters:**providers - an Iterator containing service provider objects to be registered. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if providers is null or contains a null entry.

### deregisterServiceProvider

public <T> boolean **deregisterServiceProvider**(T provider,  
 [Class](http://docs.google.com/java/lang/Class.html)<T> category)

Removes a service provider object from the given category. If the provider was not previously registered, nothing happens and false is returned. Otherwise, true is returned. If an object of the same class as provider but not equal (using ==) to provider is registered, it will not be deregistered.

If provider implements the RegisterableService interface, its onDeregistration method will be called.

**Parameters:**provider - the service provider object to be deregistered.category - the category from which to deregister the provider. **Returns:**true if the provider was previously registered in the same category category, false otherwise. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if provider is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if there is no category corresponding to category. [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if provider does not implement the class defined by category.

### deregisterServiceProvider

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

Removes a service provider object from all categories that contain it.

**Parameters:**provider - the service provider object to be deregistered. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if provider is null.

### contains

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

Returns true if provider is currently registered.

**Parameters:**provider - the service provider object to be queried. **Returns:**true if the given provider has been registered. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if provider is null.

### getServiceProviders

public <T> [Iterator](http://docs.google.com/java/util/Iterator.html)<T> **getServiceProviders**([Class](http://docs.google.com/java/lang/Class.html)<T> category,  
 boolean useOrdering)

Returns an Iterator containing all registered service providers in the given category. If useOrdering is false, the iterator will return all of the server provider objects in an arbitrary order. Otherwise, the ordering will respect any pairwise orderings that have been set. If the graph of pairwise orderings contains cycles, any providers that belong to a cycle will not be returned.

**Parameters:**category - the category to be retrieved from.useOrdering - true if pairwise orderings should be taken account in ordering the returned objects. **Returns:**an Iterator containing service provider objects from the given category, possibly in order. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if there is no category corresponding to category.

### getServiceProviders

public <T> [Iterator](http://docs.google.com/java/util/Iterator.html)<T> **getServiceProviders**([Class](http://docs.google.com/java/lang/Class.html)<T> category,  
 [ServiceRegistry.Filter](http://docs.google.com/javax/imageio/spi/ServiceRegistry.Filter.html) filter,  
 boolean useOrdering)

Returns an Iterator containing service provider objects within a given category that satisfy a criterion imposed by the supplied ServiceRegistry.Filter object's filter method.

The useOrdering argument controls the ordering of the results using the same rules as getServiceProviders(Class, boolean).

**Parameters:**category - the category to be retrieved from.filter - an instance of ServiceRegistry.Filter whose filter method will be invoked.useOrdering - true if pairwise orderings should be taken account in ordering the returned objects. **Returns:**an Iterator containing service provider objects from the given category, possibly in order. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if there is no category corresponding to category.

### getServiceProviderByClass

public <T> T **getServiceProviderByClass**([Class](http://docs.google.com/java/lang/Class.html)<T> providerClass)

Returns the currently registered service provider object that is of the given class type. At most one object of a given class is allowed to be registered at any given time. If no registered object has the desired class type, null is returned.

**Parameters:**providerClass - the Class of the desired service provider object. **Returns:**a currently registered service provider object with the desired Classtype, or null is none is present. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if providerClass is null.

### setOrdering

public <T> boolean **setOrdering**([Class](http://docs.google.com/java/lang/Class.html)<T> category,  
 T firstProvider,  
 T secondProvider)

Sets a pairwise ordering between two service provider objects within a given category. If one or both objects are not currently registered within the given category, or if the desired ordering is already set, nothing happens and false is returned. If the providers previously were ordered in the reverse direction, that ordering is removed.

The ordering will be used by the getServiceProviders methods when their useOrdering argument is true.

**Parameters:**category - a Class object indicating the category under which the preference is to be established.firstProvider - the preferred provider.secondProvider - the provider to which firstProvider is preferred. **Returns:**true if a previously unset ordering was established. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if either provider is null or they are the same object. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if there is no category corresponding to category.

### unsetOrdering

public <T> boolean **unsetOrdering**([Class](http://docs.google.com/java/lang/Class.html)<T> category,  
 T firstProvider,  
 T secondProvider)

Sets a pairwise ordering between two service provider objects within a given category. If one or both objects are not currently registered within the given category, or if no ordering is currently set between them, nothing happens and false is returned.

The ordering will be used by the getServiceProviders methods when their useOrdering argument is true.

**Parameters:**category - a Class object indicating the category under which the preference is to be disestablished.firstProvider - the formerly preferred provider.secondProvider - the provider to which firstProvider was formerly preferred. **Returns:**true if a previously set ordering was disestablished. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if either provider is null or they are the same object. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if there is no category corresponding to category.

### deregisterAll

public void **deregisterAll**([Class](http://docs.google.com/java/lang/Class.html)<?> category)

Deregisters all service provider object currently registered under the given category.

**Parameters:**category - the category to be emptied. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if there is no category corresponding to category.

### deregisterAll

public void **deregisterAll**()

Deregisters all currently registered service providers from all categories.

### finalize

public void **finalize**()  
 throws [Throwable](http://docs.google.com/java/lang/Throwable.html)

Finalizes this object prior to garbage collection. The deregisterAll method is called to deregister all currently registered service providers. This method should not be called from application code.

**Overrides:**[finalize](http://docs.google.com/java/lang/Object.html#finalize()) in class [Object](http://docs.google.com/java/lang/Object.html) **Throws:** [Throwable](http://docs.google.com/java/lang/Throwable.html) - if an error occurs during superclass finalization.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ServiceRegistry.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/imageio/spi/RegisterableService.html)   [**NEXT CLASS**](http://docs.google.com/javax/imageio/spi/ServiceRegistry.Filter.html) | [**FRAMES**](http://docs.google.com/index.html?javax/imageio/spi/ServiceRegistry.html)    [**NO FRAMES**](http://docs.google.com/ServiceRegistry.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [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).