| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/org/omg/PortableInterceptor/ORBInitInfoPackage/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/org/omg/PortableServer/CurrentPackage/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?org/omg/PortableServer/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

## Package org.omg.PortableServer

Provides classes and interfaces for making the server side of your applications portable across multivendor ORBs.

**See:**

[**Description**](#3znysh7)

| **Interface Summary** | |
| --- | --- |
| [**AdapterActivator**](http://docs.google.com/org/omg/PortableServer/AdapterActivator.html) | An adapter activator supplies a POA with the ability to create child POAs on demand, as a side-effect of receiving a request that names the child POA (or one of its children), or when find\_POA is called with an activate parameter value of TRUE. |
| [**AdapterActivatorOperations**](http://docs.google.com/org/omg/PortableServer/AdapterActivatorOperations.html) | An adapter activator supplies a POA with the ability to create child POAs on demand, as a side-effect of receiving a request that names the child POA (or one of its children), or when find\_POA is called with an activate parameter value of TRUE. |
| [**Current**](http://docs.google.com/org/omg/PortableServer/Current.html) | The PortableServer::Current interface, derived from CORBA::Current, provides method implementations with access to the identity of the object on which the method was invoked. |
| [**CurrentOperations**](http://docs.google.com/org/omg/PortableServer/CurrentOperations.html) | The PortableServer::Current interface, derived from CORBA::Current, provides method implementations with access to the identity of the object on which the method was invoked. |
| [**ID\_ASSIGNMENT\_POLICY\_ID**](http://docs.google.com/org/omg/PortableServer/ID_ASSIGNMENT_POLICY_ID.html) | org/omg/PortableServer/ID\_ASSIGNMENT\_POLICY\_ID.java . |
| [**ID\_UNIQUENESS\_POLICY\_ID**](http://docs.google.com/org/omg/PortableServer/ID_UNIQUENESS_POLICY_ID.html) | org/omg/PortableServer/ID\_UNIQUENESS\_POLICY\_ID.java . |
| [**IdAssignmentPolicy**](http://docs.google.com/org/omg/PortableServer/IdAssignmentPolicy.html) | IdAssignmentPolicy specifies whether Object Ids in the created POA are generated by the application or by the ORB. |
| [**IdAssignmentPolicyOperations**](http://docs.google.com/org/omg/PortableServer/IdAssignmentPolicyOperations.html) | IdAssignmentPolicy specifies whether Object Ids in the created POA are generated by the application or by the ORB. |
| [**IdUniquenessPolicy**](http://docs.google.com/org/omg/PortableServer/IdUniquenessPolicy.html) | The IdUniquenessPolicy specifies whether the servants activated in the created POA must have unique object i identities. |
| [**IdUniquenessPolicyOperations**](http://docs.google.com/org/omg/PortableServer/IdUniquenessPolicyOperations.html) | The IdUniquenessPolicy specifies whether the servants activated in the created POA must have unique object i identities. |
| [**IMPLICIT\_ACTIVATION\_POLICY\_ID**](http://docs.google.com/org/omg/PortableServer/IMPLICIT_ACTIVATION_POLICY_ID.html) | org/omg/PortableServer/IMPLICIT\_ACTIVATION\_POLICY\_ID.java . |
| [**ImplicitActivationPolicy**](http://docs.google.com/org/omg/PortableServer/ImplicitActivationPolicy.html) | This policy specifies whether implicit activation of servants is supported in the created POA. |
| [**ImplicitActivationPolicyOperations**](http://docs.google.com/org/omg/PortableServer/ImplicitActivationPolicyOperations.html) | This policy specifies whether implicit activation of servants is supported in the created POA. |
| [**LIFESPAN\_POLICY\_ID**](http://docs.google.com/org/omg/PortableServer/LIFESPAN_POLICY_ID.html) | org/omg/PortableServer/LIFESPAN\_POLICY\_ID.java . |
| [**LifespanPolicy**](http://docs.google.com/org/omg/PortableServer/LifespanPolicy.html) | The LifespanPolicy specifies the lifespan of the objects implemented in the created POA. |
| [**LifespanPolicyOperations**](http://docs.google.com/org/omg/PortableServer/LifespanPolicyOperations.html) | The LifespanPolicy specifies the lifespan of the objects implemented in the created POA. |
| [**POA**](http://docs.google.com/org/omg/PortableServer/POA.html) | A POA object manages the implementation of a collection of objects. |
| [**POAManager**](http://docs.google.com/org/omg/PortableServer/POAManager.html) | Each POA object has an associated POAManager object. |
| [**POAManagerOperations**](http://docs.google.com/org/omg/PortableServer/POAManagerOperations.html) | Each POA object has an associated POAManager object. |
| [**POAOperations**](http://docs.google.com/org/omg/PortableServer/POAOperations.html) | A POA object manages the implementation of a collection of objects. |
| [**REQUEST\_PROCESSING\_POLICY\_ID**](http://docs.google.com/org/omg/PortableServer/REQUEST_PROCESSING_POLICY_ID.html) | org/omg/PortableServer/REQUEST\_PROCESSING\_POLICY\_ID.java . |
| [**RequestProcessingPolicy**](http://docs.google.com/org/omg/PortableServer/RequestProcessingPolicy.html) | This policy specifies how requests are processed by the created POA. |
| [**RequestProcessingPolicyOperations**](http://docs.google.com/org/omg/PortableServer/RequestProcessingPolicyOperations.html) | This policy specifies how requests are processed by the created POA. |
| [**SERVANT\_RETENTION\_POLICY\_ID**](http://docs.google.com/org/omg/PortableServer/SERVANT_RETENTION_POLICY_ID.html) | org/omg/PortableServer/SERVANT\_RETENTION\_POLICY\_ID.java . |
| [**ServantActivator**](http://docs.google.com/org/omg/PortableServer/ServantActivator.html) | When the POA has the RETAIN policy it uses servant managers that are ServantActivators. |
| [**ServantActivatorOperations**](http://docs.google.com/org/omg/PortableServer/ServantActivatorOperations.html) | When the POA has the RETAIN policy it uses servant managers that are ServantActivators. |
| [**ServantLocator**](http://docs.google.com/org/omg/PortableServer/ServantLocator.html) | When the POA has the NON\_RETAIN policy it uses servant managers that are ServantLocators. |
| [**ServantLocatorOperations**](http://docs.google.com/org/omg/PortableServer/ServantLocatorOperations.html) | When the POA has the NON\_RETAIN policy it uses servant managers that are ServantLocators. |
| [**ServantManager**](http://docs.google.com/org/omg/PortableServer/ServantManager.html) | A servant manager supplies a POA with the ability to activate objects on demand when the POA receives a request targeted at an inactive object. |
| [**ServantManagerOperations**](http://docs.google.com/org/omg/PortableServer/ServantManagerOperations.html) | A servant manager supplies a POA with the ability to activate objects on demand when the POA receives a request targeted at an inactive object. |
| [**ServantRetentionPolicy**](http://docs.google.com/org/omg/PortableServer/ServantRetentionPolicy.html) | This policy specifies whether the created POA retains active servants in an Active Object Map. |
| [**ServantRetentionPolicyOperations**](http://docs.google.com/org/omg/PortableServer/ServantRetentionPolicyOperations.html) | This policy specifies whether the created POA retains active servants in an Active Object Map. |
| [**THREAD\_POLICY\_ID**](http://docs.google.com/org/omg/PortableServer/THREAD_POLICY_ID.html) | org/omg/PortableServer/THREAD\_POLICY\_ID.java . |
| [**ThreadPolicy**](http://docs.google.com/org/omg/PortableServer/ThreadPolicy.html) | The ThreadPolicy specifies the threading model used with the created POA. |
| [**ThreadPolicyOperations**](http://docs.google.com/org/omg/PortableServer/ThreadPolicyOperations.html) | The ThreadPolicy specifies the threading model used with the created POA. |

| **Class Summary** | |
| --- | --- |
| [**\_ServantActivatorStub**](http://docs.google.com/org/omg/PortableServer/_ServantActivatorStub.html) | When the POA has the RETAIN policy it uses servant managers that are ServantActivators. |
| [**\_ServantLocatorStub**](http://docs.google.com/org/omg/PortableServer/_ServantLocatorStub.html) | When the POA has the NON\_RETAIN policy it uses servant managers that are ServantLocators. |
| [**CurrentHelper**](http://docs.google.com/org/omg/PortableServer/CurrentHelper.html) | The PortableServer::Current interface, derived from CORBA::Current, provides method implementations with access to the identity of the object on which the method was invoked. |
| [**DynamicImplementation**](http://docs.google.com/org/omg/PortableServer/DynamicImplementation.html) | Allows dynamic handling of object invocations. |
| [**ForwardRequestHelper**](http://docs.google.com/org/omg/PortableServer/ForwardRequestHelper.html) | org/omg/PortableServer/ForwardRequestHelper.java . |
| [**IdAssignmentPolicyValue**](http://docs.google.com/org/omg/PortableServer/IdAssignmentPolicyValue.html) | The IdAssignmentPolicyValue can have the following values. |
| [**IdUniquenessPolicyValue**](http://docs.google.com/org/omg/PortableServer/IdUniquenessPolicyValue.html) | IdUniquenessPolicyValue can have the following values. |
| [**ImplicitActivationPolicyValue**](http://docs.google.com/org/omg/PortableServer/ImplicitActivationPolicyValue.html) | ImplicitActivationPolicyValue has the following semantics. |
| [**LifespanPolicyValue**](http://docs.google.com/org/omg/PortableServer/LifespanPolicyValue.html) | The LifespanPolicyValue can have the following values. |
| [**POAHelper**](http://docs.google.com/org/omg/PortableServer/POAHelper.html) | A POA object manages the implementation of a collection of objects. |
| [**RequestProcessingPolicyValue**](http://docs.google.com/org/omg/PortableServer/RequestProcessingPolicyValue.html) | The RequestProcessingPolicyValue can have the following values. |
| [**Servant**](http://docs.google.com/org/omg/PortableServer/Servant.html) | Defines the native Servant type. |
| [**ServantActivatorHelper**](http://docs.google.com/org/omg/PortableServer/ServantActivatorHelper.html) | When the POA has the RETAIN policy it uses servant managers that are ServantActivators. |
| [**ServantActivatorPOA**](http://docs.google.com/org/omg/PortableServer/ServantActivatorPOA.html) | When the POA has the RETAIN policy it uses servant managers that are ServantActivators. |
| [**ServantLocatorHelper**](http://docs.google.com/org/omg/PortableServer/ServantLocatorHelper.html) | When the POA has the NON\_RETAIN policy it uses servant managers that are ServantLocators. |
| [**ServantLocatorPOA**](http://docs.google.com/org/omg/PortableServer/ServantLocatorPOA.html) | When the POA has the NON\_RETAIN policy it uses servant managers that are ServantLocators. |
| [**ServantRetentionPolicyValue**](http://docs.google.com/org/omg/PortableServer/ServantRetentionPolicyValue.html) | ServantRetentionPolicyValue can have the following values. |
| [**ThreadPolicyValue**](http://docs.google.com/org/omg/PortableServer/ThreadPolicyValue.html) | The ThreadPolicyValue can have the following values. |

| **Exception Summary** | |
| --- | --- |
| [**ForwardRequest**](http://docs.google.com/org/omg/PortableServer/ForwardRequest.html) | org/omg/PortableServer/ForwardRequest.java . |

## Package org.omg.PortableServer Description

Provides classes and interfaces for making the server side of your applications portable across multivendor ORBs.

In Java, Portable Object Adaptor (POA)-based Dynamic Skeleton Interface (DSI) servants inherit from the standard DynamicImplementation class, which inherits from the Servant class. The native Servant type is defined by the PortableServer module for the POA. In Java, the Servant type is mapped to the Java org.omg.PortableServer.Servant class. It serves as the base class for all POA servant implementations and provides a number of methods that may be invoked by the application programmer, as well as methods which are invoked by the POA itself and may be overridden by the user to control aspects of servant behavior.

## Package Specification

For a precise list of supported sections of official OMG specifications with which the Java[tm] Platform, Standard Edition 6 complies, see [Official Specifications for CORBA support in Java[tm] SE 6](http://docs.google.com/CORBA/doc-files/compliance.html).

## POA-related Interfaces

The PortableServer module defines the following POA-related interfaces:

* POA
* POAManager
* ServantManager
* ServantActivator
* ServantLocator
* AdapterActivator
* ThreadPolicy
* LifespanPolicy
* IdUniquenessPolicy
* IdAssignmentPolicy
* ImplicitActivationPolicy
* ServantRetentionPolicy
* RequestProcessingPolicy
* Current

In addition, the POA defines the Servant native type.

### Operations classes

Each of the interfaces listed above has an associated Operations interface. The Operations interface is generated by the idlj compiler and contains the method signatures for methods defined in its associated interface. The Operations interface can be accessed by both the client and the server, while its associated interface can only be called by the client.

### Value Classes

Classes ending in the suffix PolicyValue provide the values used for the create\_POA call, which sets the policy for the POA. See the [sample code](#2et92p0) below for a demonstration. PolicyValue files include the following:

* IdAssignmentPolicyValue
* IdUniquenessPolicyValue
* ImplicitActivationPolicyValue
* LifespanPolicyValue
* RequestProcessingPolicyValue
* ServantRetentionPolicyValue
* ThreadPolicyValue

### Helper Classes

Helper classes, which are generated for all user-defined types in an OMG IDL interface, supply static methods needed to manipulate those types. There is only one method in a helper class that an application programmer uses: the narrow method. Only Java interfaces mapped from IDL interfaces will have a helper class that includes a narrow method, so in the PortableServer package, only the following classes have a narrow method:

* ForwardRequestHelper
* ServantActivatorHelper
* ServantLocatorHelper

### POA Classes

POA classes are used to implement the ServantActivator or ServantLocator.

### Exceptions

The ForwardRequest exception indicates to the ORB that it is responsible for delivering the current request and subsequent ForwardRequest requests to the object denoted in the forward\_reference member of the exception.

### Interfaces Implemented by the Application Programmer

Most of what PortableServer does is transparent to the user. The result is that programmers will use only a few of the interfaces mentioned above. The remaining interfaces will be provided by the ORB implementation. The interfaces of interest to application programmers are the following:

* AdapterActivator  
  Adapter activators are associated with POAs. An adapter activator supplies a POA with the ability to create child POAs on demand, as a side-effect of receiving a request that names the child POA (or one of its children), or when find\_POA is called with an activate parameter value of TRUE. An application server that creates all its needed POAs at the beginning of execution does not need to use or provide an adapter activator; it is necessary only for the case in which POAs need to be created during request processing.
* ServantLocator  
  When the POA has the NON\_RETAIN policy, it uses servant managers that are ServantLocators.
* ServantActivator  
  When the POA has the RETAIN policy, it uses servant managers that are ServantActivators.

## Package org.omg.PortableServer.ServantLocatorPackage

This package supplies a CookieHolder class for passing the Cookie type as an out parameter. The CookieHolder class follows exactly the same pattern as the other holder classes for basic types.

## Related Documentation

For an overview of Java IDL, please see:

* [Java IDL home page](http://docs.google.com/technotes/guides/idl/index.html).  
  Example Code  
  Example Server Code  
    
  import javax.naming.InitialContext;  
  import javax.naming.Context;  
  import javax.rmi.PortableRemoteObject ;  
  import com.sun.corba.se.impl.poa.POAORB;  
  import org.omg.PortableServer.\*;  
  import java.util.\*;  
  import org.omg.CORBA.\*;  
  import javax.rmi.CORBA.Stub;  
  import javax.rmi.CORBA.Util;  
    
    
    
  public class HelloServer {  
   public HelloServer(String[] args) {  
   try {  
   Properties p = System.getProperties();  
   // p.put("org.omg.CORBA.ORBClass", "com.sun.corba.ee.internal.POA.POAORB");  
   ORB orb = ORB.init( args, p );  
    
   POA rootPOA = (POA)orb.resolve\_initial\_references("RootPOA");  
   **Policy[] tpolicy = new Policy[3];  
   tpolicy[0] = rootPOA.create\_lifespan\_policy(  
   LifespanPolicyValue.TRANSIENT );  
   tpolicy[1] = rootPOA.create\_request\_processing\_policy(  
   RequestProcessingPolicyValue.USE\_ACTIVE\_OBJECT\_MAP\_ONLY );  
   tpolicy[2] = rootPOA.create\_servant\_retention\_policy(  
   ServantRetentionPolicyValue.RETAIN);  
   POA tpoa = rootPOA.create\_POA("MyTransientPOA", null, tpolicy);**  
    
   String ObjectId = "MyObjectId";  
   byte[] oid = ObjectId.getBytes();  
    
   org.omg.CORBA.Object obj = tpoa.create\_reference\_with\_id(oid,  
   new \_HelloImpl\_Tie().\_all\_interfaces(tpoa, oid)[0]);  
   HelloInterface helloRef = (HelloInterface)PortableRemoteObject.narrow(  
   obj, HelloInterface.class );  
    
   Context initialNamingContext = new InitialContext();  
   initialNamingContext.rebind("HelloService", helloRef);  
   System.out.println("Hello Server: Ready...");  
   orb.run();  
   } catch (Exception e) {  
   System.out.println("Trouble: " + e);  
   e.printStackTrace();  
   }   
   }  
    
    
   public static void main(String args[]) {  
   new HelloServer( args );  
   }  
  }  
    
    
    
    
    
  **Since:** 1.4

| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/org/omg/PortableInterceptor/ORBInitInfoPackage/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/org/omg/PortableServer/CurrentPackage/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?org/omg/PortableServer/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

* [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).