| | [**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/ietf/jgss/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/org/omg/CORBA_2_3/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?org/omg/CORBA/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

## Package org.omg.CORBA

Provides the mapping of the OMG CORBA APIs to the JavaTM programming language, including the class ORB, which is implemented so that a programmer can use it as a fully-functional Object Request Broker (ORB).

**See:**

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

| **Interface Summary** | |
| --- | --- |
| [**ARG\_IN**](http://docs.google.com/org/omg/CORBA/ARG_IN.html) | Signifies an "input" argument to an invocation, meaning that the argument is being passed from the client to the server. |
| [**ARG\_INOUT**](http://docs.google.com/org/omg/CORBA/ARG_INOUT.html) | Signifies an argument used for both input and output in an invocation, meaning that the argument is being passed from the client to the server and then back from the server to the client. |
| [**ARG\_OUT**](http://docs.google.com/org/omg/CORBA/ARG_OUT.html) | A constant that signifies an "output" argument to an invocation, meaning that the argument is being passed from the server to the client. |
| [**BAD\_POLICY**](http://docs.google.com/org/omg/CORBA/BAD_POLICY.html) | A PolicyErrorCode which would be filled in the PolicyError exception. |
| [**BAD\_POLICY\_TYPE**](http://docs.google.com/org/omg/CORBA/BAD_POLICY_TYPE.html) | A PolicyErrorCode which would be filled in the PolicyError exception. |
| [**BAD\_POLICY\_VALUE**](http://docs.google.com/org/omg/CORBA/BAD_POLICY_VALUE.html) | Contains the value used to indicate a policy value that is incorrect for a valid policy type in a call to the create\_policy method defined in the ORB class. |
| [**CTX\_RESTRICT\_SCOPE**](http://docs.google.com/org/omg/CORBA/CTX_RESTRICT_SCOPE.html) | A flag that can be used as the second parameter to the method Context.get\_values to restrict the search scope. |
| [**Current**](http://docs.google.com/org/omg/CORBA/Current.html) | Interfaces derived from the Current interface enable ORB and CORBA services to provide access to information (context) associated with the thread of execution in which they are running. |
| [**CurrentOperations**](http://docs.google.com/org/omg/CORBA/CurrentOperations.html) | The interface for Current. |
| [**CustomMarshal**](http://docs.google.com/org/omg/CORBA/CustomMarshal.html) | An abstract value type that is meant to be used by the ORB, not the user. |
| [**DataInputStream**](http://docs.google.com/org/omg/CORBA/DataInputStream.html) | Defines the methods used to read primitive data types from input streams for unmarshaling custom value types. |
| [**DataOutputStream**](http://docs.google.com/org/omg/CORBA/DataOutputStream.html) | Defines the methods used to write primitive data types to output streams for marshalling custom value types. |
| [**DomainManager**](http://docs.google.com/org/omg/CORBA/DomainManager.html) | Provides mechanisms for establishing and navigating relationships to superior and subordinate domains, as well as for creating and accessing policies. |
| [**DomainManagerOperations**](http://docs.google.com/org/omg/CORBA/DomainManagerOperations.html) | Provides the DomainManager with the means to access policies. |
| [**DynAny**](http://docs.google.com/org/omg/CORBA/DynAny.html) | **Deprecated.** *Use the new* [*DynAny*](http://docs.google.com/DynamicAny/DynAny.html) *instead* |
| [**DynArray**](http://docs.google.com/org/omg/CORBA/DynArray.html) | **Deprecated.** *Use the new* [*DynArray*](http://docs.google.com/DynamicAny/DynArray.html) *instead* |
| [**DynEnum**](http://docs.google.com/org/omg/CORBA/DynEnum.html) | **Deprecated.** *Use the new* [*DynEnum*](http://docs.google.com/DynamicAny/DynEnum.html) *instead* |
| [**DynFixed**](http://docs.google.com/org/omg/CORBA/DynFixed.html) | **Deprecated.** *Use the new* [*DynFixed*](http://docs.google.com/DynamicAny/DynFixed.html) *instead* |
| [**DynSequence**](http://docs.google.com/org/omg/CORBA/DynSequence.html) | **Deprecated.** *Use the new* [*DynSequence*](http://docs.google.com/DynamicAny/DynSequence.html) *instead* |
| [**DynStruct**](http://docs.google.com/org/omg/CORBA/DynStruct.html) | **Deprecated.** *Use the new* [*DynStruct*](http://docs.google.com/DynamicAny/DynStruct.html) *instead* |
| [**DynUnion**](http://docs.google.com/org/omg/CORBA/DynUnion.html) | **Deprecated.** *Use the new* [*DynUnion*](http://docs.google.com/DynamicAny/DynUnion.html) *instead* |
| [**DynValue**](http://docs.google.com/org/omg/CORBA/DynValue.html) | **Deprecated.** *Use the new* [*DynValue*](http://docs.google.com/DynamicAny/DynValue.html) *instead* |
| [**IDLType**](http://docs.google.com/org/omg/CORBA/IDLType.html) | An abstract interface inherited by all Interface Repository (IR) objects that represent OMG IDL types. |
| [**IDLTypeOperations**](http://docs.google.com/org/omg/CORBA/IDLTypeOperations.html) | This interface must be implemented by all IDLType objects. |
| [**IRObject**](http://docs.google.com/org/omg/CORBA/IRObject.html) | An IRObject IDL interface represents the most generic interface from which all other Interface Repository interfaces are derived, even the Repository itself. |
| [**IRObjectOperations**](http://docs.google.com/org/omg/CORBA/IRObjectOperations.html) | This is the Operations interface for the mapping from IRObject. |
| [**Object**](http://docs.google.com/org/omg/CORBA/Object.html) | The definition for a CORBA object reference. |
| [**OMGVMCID**](http://docs.google.com/org/omg/CORBA/OMGVMCID.html) | The vendor minor code ID reserved for OMG. |
| [**Policy**](http://docs.google.com/org/omg/CORBA/Policy.html) | Interfaces derived from the Policy interface allow an ORB or CORBA service access to certain choices that affect its operation. |
| [**PolicyOperations**](http://docs.google.com/org/omg/CORBA/PolicyOperations.html) | Provides the operations for a Policy object. |
| [**PRIVATE\_MEMBER**](http://docs.google.com/org/omg/CORBA/PRIVATE_MEMBER.html) | Constant to define a private member in the ValueMember class. |
| [**PUBLIC\_MEMBER**](http://docs.google.com/org/omg/CORBA/PUBLIC_MEMBER.html) | Constant to define a public member in the ValueMember class. |
| [**UNSUPPORTED\_POLICY**](http://docs.google.com/org/omg/CORBA/UNSUPPORTED_POLICY.html) | One of the PolicyErrorCodes which would be filled if the requested Policy is understood to be valid by the ORB, but is not currently supported. |
| [**UNSUPPORTED\_POLICY\_VALUE**](http://docs.google.com/org/omg/CORBA/UNSUPPORTED_POLICY_VALUE.html) | A PolicyErrorCode which would be filled if the value requested for the Policy is of a valid type and within the valid range for that type, but this valid value is not currently supported. |
| [**VM\_ABSTRACT**](http://docs.google.com/org/omg/CORBA/VM_ABSTRACT.html) | Defines the code used to represent an Abstract interface in a typecode. |
| [**VM\_CUSTOM**](http://docs.google.com/org/omg/CORBA/VM_CUSTOM.html) | Defines the code used to represent a custom marshalled value type in a typecode. |
| [**VM\_NONE**](http://docs.google.com/org/omg/CORBA/VM_NONE.html) | Defines the code used to represent the one of the values of a value type in a typecode. |
| [**VM\_TRUNCATABLE**](http://docs.google.com/org/omg/CORBA/VM_TRUNCATABLE.html) | Defines the code used to represent a truncatable value type in a typecode. |

| **Class Summary** | |
| --- | --- |
| [**\_IDLTypeStub**](http://docs.google.com/org/omg/CORBA/_IDLTypeStub.html) | The Stub for IDLType. |
| [**\_PolicyStub**](http://docs.google.com/org/omg/CORBA/_PolicyStub.html) | The Stub for Policy. |
| [**Any**](http://docs.google.com/org/omg/CORBA/Any.html) | Serves as a container for any data that can be described in IDL or for any IDL primitive type. |
| [**AnyHolder**](http://docs.google.com/org/omg/CORBA/AnyHolder.html) | The Holder for Any. |
| [**AnySeqHelper**](http://docs.google.com/org/omg/CORBA/AnySeqHelper.html) | The Helper for AnySeq. |
| [**AnySeqHolder**](http://docs.google.com/org/omg/CORBA/AnySeqHolder.html) | The Holder for AnySeq. |
| [**BooleanHolder**](http://docs.google.com/org/omg/CORBA/BooleanHolder.html) | The Holder for Boolean. |
| [**BooleanSeqHelper**](http://docs.google.com/org/omg/CORBA/BooleanSeqHelper.html) | The Helper for BooleanSeq. |
| [**BooleanSeqHolder**](http://docs.google.com/org/omg/CORBA/BooleanSeqHolder.html) | The Holder for BooleanSeq. |
| [**ByteHolder**](http://docs.google.com/org/omg/CORBA/ByteHolder.html) | The Holder for Byte. |
| [**CharHolder**](http://docs.google.com/org/omg/CORBA/CharHolder.html) | The Holder for Char. |
| [**CharSeqHelper**](http://docs.google.com/org/omg/CORBA/CharSeqHelper.html) | The Helper for CharSeq. |
| [**CharSeqHolder**](http://docs.google.com/org/omg/CORBA/CharSeqHolder.html) | The Holder for CharSeq. |
| [**CompletionStatus**](http://docs.google.com/org/omg/CORBA/CompletionStatus.html) | An object that indicates whether a method had completed running when a SystemException was thrown. |
| [**CompletionStatusHelper**](http://docs.google.com/org/omg/CORBA/CompletionStatusHelper.html) | The Helper for CompletionStatus. |
| [**Context**](http://docs.google.com/org/omg/CORBA/Context.html) | An object used in Request operations to specify the context object in which context strings must be resolved before being sent along with the request invocation. |
| [**ContextList**](http://docs.google.com/org/omg/CORBA/ContextList.html) | An object containing a modifiable list of String objects that represent property names. |
| [**CurrentHelper**](http://docs.google.com/org/omg/CORBA/CurrentHelper.html) | The Helper for Current. |
| [**CurrentHolder**](http://docs.google.com/org/omg/CORBA/CurrentHolder.html) | The Holder for Current. |
| [**DefinitionKind**](http://docs.google.com/org/omg/CORBA/DefinitionKind.html) | The class that provides the constants used to identify the type of an Interface Repository object. |
| [**DefinitionKindHelper**](http://docs.google.com/org/omg/CORBA/DefinitionKindHelper.html) | The Helper for DefinitionKind. |
| [**DoubleHolder**](http://docs.google.com/org/omg/CORBA/DoubleHolder.html) | The Holder for Double. |
| [**DoubleSeqHelper**](http://docs.google.com/org/omg/CORBA/DoubleSeqHelper.html) | The Helper for DoubleSeq. |
| [**DoubleSeqHolder**](http://docs.google.com/org/omg/CORBA/DoubleSeqHolder.html) | The Holder for DoubleSeq. |
| [**DynamicImplementation**](http://docs.google.com/org/omg/CORBA/DynamicImplementation.html) | **Deprecated.** *org.omg.CORBA.DynamicImplementation* |
| [**Environment**](http://docs.google.com/org/omg/CORBA/Environment.html) | A container (holder) for an exception that is used in Request operations to make exceptions available to the client. |
| [**ExceptionList**](http://docs.google.com/org/omg/CORBA/ExceptionList.html) | An object used in Request operations to describe the exceptions that can be thrown by a method. |
| [**FieldNameHelper**](http://docs.google.com/org/omg/CORBA/FieldNameHelper.html) | The Helper for FieldName. |
| [**FixedHolder**](http://docs.google.com/org/omg/CORBA/FixedHolder.html) | The Holder for Fixed. |
| [**FloatHolder**](http://docs.google.com/org/omg/CORBA/FloatHolder.html) | The Holder for Float. |
| [**FloatSeqHelper**](http://docs.google.com/org/omg/CORBA/FloatSeqHelper.html) | The Helper for FloatSeq. |
| [**FloatSeqHolder**](http://docs.google.com/org/omg/CORBA/FloatSeqHolder.html) | The Holder for FloatSeq. |
| [**IdentifierHelper**](http://docs.google.com/org/omg/CORBA/IdentifierHelper.html) | The Helper for Identifier. |
| [**IDLTypeHelper**](http://docs.google.com/org/omg/CORBA/IDLTypeHelper.html) | The Helper for IDLType. |
| [**IntHolder**](http://docs.google.com/org/omg/CORBA/IntHolder.html) | The Holder for Int. |
| [**LocalObject**](http://docs.google.com/org/omg/CORBA/LocalObject.html) | Used as a base class for implementation of a local IDL interface in the Java language mapping. |
| [**LongHolder**](http://docs.google.com/org/omg/CORBA/LongHolder.html) | The Holder for Long. |
| [**LongLongSeqHelper**](http://docs.google.com/org/omg/CORBA/LongLongSeqHelper.html) | The Helper for LongLongSeq. |
| [**LongLongSeqHolder**](http://docs.google.com/org/omg/CORBA/LongLongSeqHolder.html) | The Holder for LongLongSeq. |
| [**LongSeqHelper**](http://docs.google.com/org/omg/CORBA/LongSeqHelper.html) | The Helper for LongSeqHelper. |
| [**LongSeqHolder**](http://docs.google.com/org/omg/CORBA/LongSeqHolder.html) | The Holder for LongSeq. |
| [**NamedValue**](http://docs.google.com/org/omg/CORBA/NamedValue.html) | An object used in the DII and DSI to describe arguments and return values. |
| [**NameValuePair**](http://docs.google.com/org/omg/CORBA/NameValuePair.html) | Associates a name with a value that is an attribute of an IDL struct, and is used in the DynStruct APIs. |
| [**NameValuePairHelper**](http://docs.google.com/org/omg/CORBA/NameValuePairHelper.html) | The Helper for NameValuePair. |
| [**NVList**](http://docs.google.com/org/omg/CORBA/NVList.html) | A modifiable list containing NamedValue objects. |
| [**ObjectHelper**](http://docs.google.com/org/omg/CORBA/ObjectHelper.html) |  |
| [**ObjectHolder**](http://docs.google.com/org/omg/CORBA/ObjectHolder.html) | The Holder for Object. |
| [**OctetSeqHelper**](http://docs.google.com/org/omg/CORBA/OctetSeqHelper.html) | The Helper for OctetSeq. |
| [**OctetSeqHolder**](http://docs.google.com/org/omg/CORBA/OctetSeqHolder.html) | The Holder for OctetSeq. |
| [**ORB**](http://docs.google.com/org/omg/CORBA/ORB.html) | A class providing APIs for the CORBA Object Request Broker features. |
| [**ParameterMode**](http://docs.google.com/org/omg/CORBA/ParameterMode.html) | Enumeration of parameter modes for Parameter. |
| [**ParameterModeHelper**](http://docs.google.com/org/omg/CORBA/ParameterModeHelper.html) | Enumeration of parameter modes for Parameter. |
| [**ParameterModeHolder**](http://docs.google.com/org/omg/CORBA/ParameterModeHolder.html) | Enumeration of parameter modes for Parameter. |
| [**PolicyErrorCodeHelper**](http://docs.google.com/org/omg/CORBA/PolicyErrorCodeHelper.html) | Encapsulates a reason a Policy may be invalid. |
| [**PolicyErrorHelper**](http://docs.google.com/org/omg/CORBA/PolicyErrorHelper.html) | Thrown to indicate problems with parameter values passed to the ORB.create\_policy operation. |
| [**PolicyErrorHolder**](http://docs.google.com/org/omg/CORBA/PolicyErrorHolder.html) | Thrown to indicate problems with parameter values passed to the ORB.create\_policy operation. |
| [**PolicyHelper**](http://docs.google.com/org/omg/CORBA/PolicyHelper.html) | The Helper for Policy. |
| [**PolicyHolder**](http://docs.google.com/org/omg/CORBA/PolicyHolder.html) | The Holder for Policy. |
| [**PolicyListHelper**](http://docs.google.com/org/omg/CORBA/PolicyListHelper.html) | The Helper for PolicyList. |
| [**PolicyListHolder**](http://docs.google.com/org/omg/CORBA/PolicyListHolder.html) | The Holder for PolicyList. |
| [**PolicyTypeHelper**](http://docs.google.com/org/omg/CORBA/PolicyTypeHelper.html) | The Helper for PolicyType. |
| [**Principal**](http://docs.google.com/org/omg/CORBA/Principal.html) | **Deprecated.** *Deprecated by CORBA 2.2.* |
| [**PrincipalHolder**](http://docs.google.com/org/omg/CORBA/PrincipalHolder.html) | **Deprecated.** *Deprecated by CORBA 2.2.* |
| [**RepositoryIdHelper**](http://docs.google.com/org/omg/CORBA/RepositoryIdHelper.html) | The Helper for RepositoryId. |
| [**Request**](http://docs.google.com/org/omg/CORBA/Request.html) | An object containing the information necessary for invoking a method. |
| [**ServerRequest**](http://docs.google.com/org/omg/CORBA/ServerRequest.html) | An object that captures the explicit state of a request for the Dynamic Skeleton Interface (DSI). |
| [**ServiceDetail**](http://docs.google.com/org/omg/CORBA/ServiceDetail.html) | An object that represents an ORB service: its service\_detail\_type field contains the type of the ORB service, and its service\_detail field contains a description of the ORB service. |
| [**ServiceDetailHelper**](http://docs.google.com/org/omg/CORBA/ServiceDetailHelper.html) |  |
| [**ServiceInformation**](http://docs.google.com/org/omg/CORBA/ServiceInformation.html) | An IDL struct in the CORBA module that stores information about a CORBA service available in the ORB implementation and is obtained from the ORB.get\_service\_information method. |
| [**ServiceInformationHelper**](http://docs.google.com/org/omg/CORBA/ServiceInformationHelper.html) |  |
| [**ServiceInformationHolder**](http://docs.google.com/org/omg/CORBA/ServiceInformationHolder.html) | The Holder for ServiceInformation. |
| [**SetOverrideType**](http://docs.google.com/org/omg/CORBA/SetOverrideType.html) | The mapping of a CORBA enum tagging SET\_OVERRIDE and ADD\_OVERRIDE, which indicate whether policies should replace the existing policies of an Object or be added to them. |
| [**SetOverrideTypeHelper**](http://docs.google.com/org/omg/CORBA/SetOverrideTypeHelper.html) | The Helper for SetOverrideType. |
| [**ShortHolder**](http://docs.google.com/org/omg/CORBA/ShortHolder.html) | The Holder for Short. |
| [**ShortSeqHelper**](http://docs.google.com/org/omg/CORBA/ShortSeqHelper.html) | The Helper for ShortSeqHelper. |
| [**ShortSeqHolder**](http://docs.google.com/org/omg/CORBA/ShortSeqHolder.html) | The Holder for ShortSeq. |
| [**StringHolder**](http://docs.google.com/org/omg/CORBA/StringHolder.html) | The Holder for String. |
| [**StringSeqHelper**](http://docs.google.com/org/omg/CORBA/StringSeqHelper.html) | An array of Strings |
| [**StringSeqHolder**](http://docs.google.com/org/omg/CORBA/StringSeqHolder.html) | An array of Strings |
| [**StringValueHelper**](http://docs.google.com/org/omg/CORBA/StringValueHelper.html) | The Helper for StringValue. |
| [**StructMember**](http://docs.google.com/org/omg/CORBA/StructMember.html) | Describes a member of an IDL struct in the Interface Repository, including the name of the struct member, the type of the struct member, and the typedef that represents the IDL type of the struct member described the struct member object. |
| [**StructMemberHelper**](http://docs.google.com/org/omg/CORBA/StructMemberHelper.html) | The Helper for StructMember. |
| [**TCKind**](http://docs.google.com/org/omg/CORBA/TCKind.html) | The Java mapping of the IDL enum TCKind, which specifies the kind of a TypeCode object. |
| [**TypeCode**](http://docs.google.com/org/omg/CORBA/TypeCode.html) | A container for information about a specific CORBA data type. |
| [**TypeCodeHolder**](http://docs.google.com/org/omg/CORBA/TypeCodeHolder.html) | The Holder for TypeCode. |
| [**ULongLongSeqHelper**](http://docs.google.com/org/omg/CORBA/ULongLongSeqHelper.html) | The Helper for ULongLongSeq. |
| [**ULongLongSeqHolder**](http://docs.google.com/org/omg/CORBA/ULongLongSeqHolder.html) | The Holder for ULongLongSeq. |
| [**ULongSeqHelper**](http://docs.google.com/org/omg/CORBA/ULongSeqHelper.html) | The Helper for ULongSeq. |
| [**ULongSeqHolder**](http://docs.google.com/org/omg/CORBA/ULongSeqHolder.html) | The Holder for ULongSeq. |
| [**UnionMember**](http://docs.google.com/org/omg/CORBA/UnionMember.html) | A description in the Interface Repository of a member of an IDL union. |
| [**UnionMemberHelper**](http://docs.google.com/org/omg/CORBA/UnionMemberHelper.html) | The Helper for UnionMember. |
| [**UnknownUserExceptionHelper**](http://docs.google.com/org/omg/CORBA/UnknownUserExceptionHelper.html) | The Helper for UnknownUserException. |
| [**UnknownUserExceptionHolder**](http://docs.google.com/org/omg/CORBA/UnknownUserExceptionHolder.html) | The Holder for UnknownUserException. |
| [**UShortSeqHelper**](http://docs.google.com/org/omg/CORBA/UShortSeqHelper.html) | The Helper for UShortSeq. |
| [**UShortSeqHolder**](http://docs.google.com/org/omg/CORBA/UShortSeqHolder.html) | The Holder for UShortSeq. |
| [**ValueBaseHelper**](http://docs.google.com/org/omg/CORBA/ValueBaseHelper.html) |  |
| [**ValueBaseHolder**](http://docs.google.com/org/omg/CORBA/ValueBaseHolder.html) | The Holder for ValueBase. |
| [**ValueMember**](http://docs.google.com/org/omg/CORBA/ValueMember.html) | A description in the Interface Repository of a member of a value object. |
| [**ValueMemberHelper**](http://docs.google.com/org/omg/CORBA/ValueMemberHelper.html) | The Helper for ValueMember. |
| [**VersionSpecHelper**](http://docs.google.com/org/omg/CORBA/VersionSpecHelper.html) | The Helper for VersionSpec. |
| [**VisibilityHelper**](http://docs.google.com/org/omg/CORBA/VisibilityHelper.html) | The Helper for Visibility. |
| [**WCharSeqHelper**](http://docs.google.com/org/omg/CORBA/WCharSeqHelper.html) | The Helper for WCharSeq. |
| [**WCharSeqHolder**](http://docs.google.com/org/omg/CORBA/WCharSeqHolder.html) | The Holder for WCharSeq. |
| [**WrongTransactionHelper**](http://docs.google.com/org/omg/CORBA/WrongTransactionHelper.html) | The Helper for WrongTransaction. |
| [**WrongTransactionHolder**](http://docs.google.com/org/omg/CORBA/WrongTransactionHolder.html) | The Holder for WrongTransaction. |
| [**WStringSeqHelper**](http://docs.google.com/org/omg/CORBA/WStringSeqHelper.html) | An array of WStrings |
| [**WStringSeqHolder**](http://docs.google.com/org/omg/CORBA/WStringSeqHolder.html) | An array of WStrings |
| [**WStringValueHelper**](http://docs.google.com/org/omg/CORBA/WStringValueHelper.html) | org/omg/CORBA/WStringValueHelper.java Generated by the IDL-to-Java compiler (portable), version "3.0" from orb.idl 31 May 1999 22:27:30 o'clock GMT+00:00 The class definition has been modified to conform to the following OMG specifications : ORB core as defined by CORBA 2.3.1 ([formal/99-10-07](http://cgi.omg.org/cgi-bin/doc?formal/99-10-07)) IDL/Java Language Mapping as defined in [ptc/00-01-08](http://cgi.omg.org/cgi-bin/doc?ptc/00-01-08) |

| **Exception Summary** | |
| --- | --- |
| [**ACTIVITY\_COMPLETED**](http://docs.google.com/org/omg/CORBA/ACTIVITY_COMPLETED.html) | The ACTIVITY\_COMPLETED system exception may be raised on any method for which Activity context is accessed. |
| [**ACTIVITY\_REQUIRED**](http://docs.google.com/org/omg/CORBA/ACTIVITY_REQUIRED.html) | The ACTIVITY\_REQUIRED system exception may be raised on any method for which an Activity context is required. |
| [**BAD\_CONTEXT**](http://docs.google.com/org/omg/CORBA/BAD_CONTEXT.html) | Exception thrown when an operation is invoked by a client but the passed context does not contain the context values required by the operation. |
| [**BAD\_INV\_ORDER**](http://docs.google.com/org/omg/CORBA/BAD_INV_ORDER.html) | This exception indicates that the caller has invoked operations in the wrong order. |
| [**BAD\_OPERATION**](http://docs.google.com/org/omg/CORBA/BAD_OPERATION.html) | Exception thrown when an object reference denotes an existing object, but that the object does not support the operation that was invoked. |
| [**BAD\_PARAM**](http://docs.google.com/org/omg/CORBA/BAD_PARAM.html) | Exception thrown when a parameter passed to a call is out of range or otherwise considered illegal. |
| [**BAD\_QOS**](http://docs.google.com/org/omg/CORBA/BAD_QOS.html) | The BAD\_QOS exception is raised whenever an object cannot support the quality of service required by an invocation parameter that has a quality of service semantics associated with it. |
| [**BAD\_TYPECODE**](http://docs.google.com/org/omg/CORBA/BAD_TYPECODE.html) | Exception thrown when the ORB has encountered a malformed type code (for example, a type code with an invalid TCKind value). |
| [**Bounds**](http://docs.google.com/org/omg/CORBA/Bounds.html) | A user exception thrown when a parameter is not within the legal bounds for the object that a method is trying to access. |
| [**CODESET\_INCOMPATIBLE**](http://docs.google.com/org/omg/CORBA/CODESET_INCOMPATIBLE.html) | This exception is raised whenever meaningful communication is not possible between client and server native code sets. |
| [**COMM\_FAILURE**](http://docs.google.com/org/omg/CORBA/COMM_FAILURE.html) | This exception is raised if communication is lost while an operation is in progress, after the request was sent by the client, but before the reply from the server has been returned to the client. |
| [**DATA\_CONVERSION**](http://docs.google.com/org/omg/CORBA/DATA_CONVERSION.html) | This exception is raised if an ORB cannot convert the representation of data as marshaled into its native representation or vice-versa. |
| [**FREE\_MEM**](http://docs.google.com/org/omg/CORBA/FREE_MEM.html) | Exception thrown when the ORB failed in an attempt to free dynamic memory, for example because of heap corruption or memory segments being locked. |
| [**IMP\_LIMIT**](http://docs.google.com/org/omg/CORBA/IMP_LIMIT.html) | This exception indicates that an implementation limit was exceeded in the ORB run time. |
| [**INITIALIZE**](http://docs.google.com/org/omg/CORBA/INITIALIZE.html) | Exception thrown when an ORB has encountered a failure during its initialization, such as failure to acquire networking resources or detecting a configuration error. |
| [**INTERNAL**](http://docs.google.com/org/omg/CORBA/INTERNAL.html) | This exception indicates an internal failure in an ORB, for example, if an ORB has detected corruption of its internal data structures. |
| [**INTF\_REPOS**](http://docs.google.com/org/omg/CORBA/INTF_REPOS.html) | Exception raised when an ORB cannot reach the interface repository, or some other failure relating to the interface repository is detected. |
| [**INV\_FLAG**](http://docs.google.com/org/omg/CORBA/INV_FLAG.html) | Exception thrown when an invalid flag was passed to an operation (for example, when creating a DII request). |
| [**INV\_IDENT**](http://docs.google.com/org/omg/CORBA/INV_IDENT.html) | This exception indicates that an IDL identifier is syntactically invalid. |
| [**INV\_OBJREF**](http://docs.google.com/org/omg/CORBA/INV_OBJREF.html) | This exception indicates that an object reference is internally malformed. |
| [**INV\_POLICY**](http://docs.google.com/org/omg/CORBA/INV_POLICY.html) | Standard exception thrown when an invocation cannot be made because of an incompatibility between Policy overrides that apply to the particular invocation. |
| [**INVALID\_ACTIVITY**](http://docs.google.com/org/omg/CORBA/INVALID_ACTIVITY.html) | The INVALID\_ACTIVITY system exception may be raised on the Activity or Transaction services' resume methods if a transaction or Activity is resumed in a context different to that from which it was suspended. |
| [**INVALID\_TRANSACTION**](http://docs.google.com/org/omg/CORBA/INVALID_TRANSACTION.html) | Exception thrown when the request carried an invalid transaction context. |
| [**MARSHAL**](http://docs.google.com/org/omg/CORBA/MARSHAL.html) | A request or reply from the network is structurally invalid. |
| [**NO\_IMPLEMENT**](http://docs.google.com/org/omg/CORBA/NO_IMPLEMENT.html) | This exception indicates that even though the operation that was invoked exists (it has an IDL definition), no implementation for that operation exists. |
| [**NO\_MEMORY**](http://docs.google.com/org/omg/CORBA/NO_MEMORY.html) | Exception thrown when the ORB run time has run out of memory. |
| [**NO\_PERMISSION**](http://docs.google.com/org/omg/CORBA/NO_PERMISSION.html) | Exception thrown when an invocation failed because the caller has insufficient privileges. |
| [**NO\_RESOURCES**](http://docs.google.com/org/omg/CORBA/NO_RESOURCES.html) | Exception thrown when the ORB has encountered some general resource limitation. |
| [**NO\_RESPONSE**](http://docs.google.com/org/omg/CORBA/NO_RESPONSE.html) | This exception is raised if a client attempts to retrieve the result of a deferred synchronous call, but the response for the request is not yet available. |
| [**OBJ\_ADAPTER**](http://docs.google.com/org/omg/CORBA/OBJ_ADAPTER.html) | This exception typically indicates an administrative mismatch, for example, a server may have made an attempt to register itself with an implementation repository under a name that is already in use, or is unknown to the repository. |
| [**OBJECT\_NOT\_EXIST**](http://docs.google.com/org/omg/CORBA/OBJECT_NOT_EXIST.html) | Exception raised whenever an invocation on a deleted object was performed. |
| [**PERSIST\_STORE**](http://docs.google.com/org/omg/CORBA/PERSIST_STORE.html) | This exception indicates a persistent storage failure, for example, failure to establish a database connection or corruption of a database. |
| [**PolicyError**](http://docs.google.com/org/omg/CORBA/PolicyError.html) | A user exception thrown when a policy error occurs. |
| [**REBIND**](http://docs.google.com/org/omg/CORBA/REBIND.html) | REBIND is raised when the current effective RebindPolicy, has a value of NO\_REBIND or NO\_RECONNECT and an invocation on a bound object reference results in a LocateReply message with status OBJECT\_FORWARD or a Reply message with status LOCATION\_FORWARD. |
| [**SystemException**](http://docs.google.com/org/omg/CORBA/SystemException.html) | The root class for all CORBA standard exceptions. |
| [**TIMEOUT**](http://docs.google.com/org/omg/CORBA/TIMEOUT.html) | TIMEOUT is raised when no delivery has been made and the specified time-to-live period has been exceeded. |
| [**TRANSACTION\_MODE**](http://docs.google.com/org/omg/CORBA/TRANSACTION_MODE.html) | The CORBA TRANSACTION\_MODE exception is thrown by the client ORB if it detects a mismatch between the InvocationPolicy in the IOR and the chosen invocation path (i.e, direct or routed invocation). |
| [**TRANSACTION\_REQUIRED**](http://docs.google.com/org/omg/CORBA/TRANSACTION_REQUIRED.html) | Exception indicates that the request carried a null transaction context, but an active transaction is required. |
| [**TRANSACTION\_ROLLEDBACK**](http://docs.google.com/org/omg/CORBA/TRANSACTION_ROLLEDBACK.html) | Exception thrown when the transaction associated with the request has already been rolled back or marked to roll back. |
| [**TRANSACTION\_UNAVAILABLE**](http://docs.google.com/org/omg/CORBA/TRANSACTION_UNAVAILABLE.html) | The CORBA TRANSACTION\_UNAVAILABLE exception is thrown by the ORB when it cannot process a transaction service context because its connection to the Transaction Service has been abnormally terminated. |
| [**TRANSIENT**](http://docs.google.com/org/omg/CORBA/TRANSIENT.html) | Exception thrown when the ORB attempted to reach an object and failed. |
| [**UNKNOWN**](http://docs.google.com/org/omg/CORBA/UNKNOWN.html) | This exception is raised if an operation implementation throws a non-CORBA exception (such as an exception specific to the implementation's programming language), or if an operation raises a user exception that does not appear in the operation's raises expression. |
| [**UnknownUserException**](http://docs.google.com/org/omg/CORBA/UnknownUserException.html) | A class that contains user exceptions returned by the server. |
| [**UserException**](http://docs.google.com/org/omg/CORBA/UserException.html) | The root class for CORBA IDL-defined user exceptions. |
| [**WrongTransaction**](http://docs.google.com/org/omg/CORBA/WrongTransaction.html) | The CORBA WrongTransaction user-defined exception. |

## Package org.omg.CORBA Description

Provides the mapping of the OMG CORBA APIs to the JavaTM programming language, including the class ORB, which is implemented so that a programmer can use it as a fully-functional Object Request Broker (ORB).

For a precise list of supported sections of official CORBA 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/doc-files/compliance.html).

# General Information

The information in this section is information relevant to someone who compiles Interface Definition Language (IDL) files and uses the ORB to write clients and servers.

The classes and interfaces described in this section can be put into four groups: ORB classes, Exceptions, Helper classes, and Holder classes.

## The ORB Class

An ORB handles (or brokers) method invocations between a client and the method's implementation on a server. Because the client and server may be anywhere on a network, and because the invocation and implementation may be written in different programming languages, an ORB does a great deal of work behind the scenes to accomplish this communication.

Most of what an ORB does is completely transparent to the user, and a major portion of the CORBA package consists of classes used by the ORB behind the scenes. The result is that most programmers will use only a small part of this package directly. In fact, most programmers will use only a few methods from the ORB class, some exceptions, and occasionally, a holder class.

### ORB Methods

Before an application can enter the CORBA environment, it must first:

* Be initialized into the ORB and possibly the object adapter (POA) environments.
* Get references to ORB object (for use in future ORB operations) and perhaps other objects (including the root POA or some Object Adapter objects).

The following operations are provided to initialize applications and obtain the appropriate object references:

* Operations providing access to the ORB, which are discussed in this section.
* Operations providing access to Object Adapters, Interface Repository, Naming Service, and other Object Services. These operations are described in [*Other Classes*](#3dy6vkm).

When an application requires a CORBA environment it needs a mechanism to get an ORB object reference and possibly an OA object reference (such as the root POA). This serves two purposes. First, it initializes an application into the ORB and OA environments. Second, it returns the ORB object reference and the OA object reference to the application for use in future ORB and OA operations.

In order to obtain an ORB object reference, applications call the ORB.init operation. The parameters to the call can comprise an identifier for the ORB for which the object reference is required, and an arg\_list, which is used to allow environment-specific data to be passed into the call.

These are the ORB methods that provide access to the ORB:

* init()
* init(String [] args, Properties props)
* init(Applet app, Properties props)

Using the init() method without parameters initiates a singleton ORB, which can only give typecode creation anys needed in code generated in Helper classes by idlj.

Applications require a portable means by which to obtain their initial object references. References are required for the root POA, POA Current, Interface Repository, and various Object Services instances. The functionality required by the application is similar to that provided by the Naming Service. However, the OMG does not want to mandate that the Naming Service be made available to all applications in order that they may be portably initialized. Consequently, the operations shown in this section provide a simplified, local version of the Naming Service that applications can use to obtain a small, defined set of object references which are essential to its operation. Because only a small well-defined set of objects are expected with this mechanism, the naming context can be flattened to be a single-level name space. This simplification results in only two operations being defined to achieve the functionality required.

Initial references are obtained via two operations provided in the ORB object interface, providing facilities to list and resolve initial object references. These are:

* resolve\_initial\_references(String name)
* list\_initial\_services()
* register\_initial\_reference(String id, org.omg.CORBA.Object obj)

An example that uses some of these methods is  [*Getting Started with Java IDL*](http://docs.google.com/technotes/guides/idl/GShome.html).

## Exceptions

Exceptions in Java IDL are similar to those in any code written in the Java programming language. If a method is defined to throw an exception, then any code using that method must have a try/catch block and handle that exception when it is thrown.

The documentation on [*Java IDL exceptions*](http://docs.google.com/technotes/guides/idl/jidlExceptions.html) has more information and explains the difference between system exceptions and user-defined exceptions.

The following is a list of the system exceptions (which are unchecked exceptions inheriting through  [org.omg.CORBA.SystemException](http://docs.google.com/SystemException.html) from java.lang.RuntimeException) that are defined in the package org.omg.CORBA:

        BAD\_CONTEXT  
        BAD\_INV\_ORDER  
        BAD\_OPERATION  
        BAD\_PARAM  
        BAD\_TYPECODE  
        COMM\_FAILURE  
        DATA\_CONVERSION  
        FREE\_MEM  
        IMP\_LIMIT  
        INITIALIZE  
        INTERNAL  
        INTF\_REPOS  
        INVALID\_TRANSACTION  
        INV\_FLAG  
        INV\_IDENT  
        INV\_OBJREF  
        INV\_POLICY  
        MARSHAL  
        [NO\_IMPLEMENT](#4d34og8)  
        NO\_MEMORY  
        NO\_PERMISSION  
        NO\_RESOURCES  
        NO\_RESPONSE  
        OBJECT\_NOT\_EXIST  
        OBJ\_ADAPTER  
        PERSIST\_STORE  
        TRANSACTION\_REQUIRED  
        TRANSACTION\_ROLLEDBACK  
        TRANSIENT  
        UNKNOWN

The following is a list of user-defined exceptions defined in the package org.omg.CORBA.

        Bounds  
        UnknownUserException  
        WrongTransaction   
        PolicyError

## Subpackages

There are some packages inside the CORBA package with "Package" as part of their names. These packages are generally quite small because all they do is provide exceptions or classes for use by interfaces and classes in the CORBA package.

For example, the package  [org.omg.CORBA.TypeCodePackage](http://docs.google.com/TypeCodePackage/package-summary.html) contains two exceptions thrown by methods in the class TypeCode. These exceptions are:

* BadKind
* Bounds

The package  [org.omg.CORBA.ORBPackage](http://docs.google.com/ORBPackage/package-summary.html) contains two exceptions:

* InvalidName
* InconsistentTypeCode

Another package that is a subpackage of CORBA is the [portable](http://docs.google.com/portable/package-summary.html) package. It provides a set of ORB APIs that makes it possible for code generated by one vendor's IDL compiler to run on another vendor's ORB.

## Holder classes

Support for out and inout parameter passing modes requires the use of additional [*holder classes*](http://docs.google.com/doc-files/generatedfiles.html#holder). Because the Java programming language does not support out or inout parameters, holder classes are needed as a means of passing a parameter that can be modified. To support portable stubs and skeletons, holder classes also implement the [org.omg.CORBA.portable.Streamable](http://docs.google.com/portable/Streamable.html) interface.

Holder classes are named by appending "Holder" to the name of the type. The name of the type refers to its name in the Java programming language. For example, a holder class for the interface named Account in the Java programming language would be named AccountHolder.

Holder classes are available for all of the basic IDL datatypes in the org.omg.CORBA package. So, for example, there are already-defined classes for LongHolder, ShortHolder, FloatHolder, and so on. Classes are also generated for all named user-defined IDL types except those defined by typedefs. (Note that in this context user defined includes types that are defined in OMG specifications such as those for the Interface Repository, and other OMG services.)

Each holder class has:

* a constructor from an instance
* a default constructor
* a public instance member, value which is the typed value.
* a method for reading an input stream and assigning the contents to the type's value field
* a method for writing the value of the value field to an output stream
* a method for getting the typecode of the type

The default constructor sets the value field to the default value for the type as defined by the Java language:

* false for boolean
* 0 for numeric and char types
* null for strings and object references

As an example, if the interface Account, defined in OMG IDL, were mapped to the Java programming language, the following holder class would be generated:

public final class AccountHolder implements   
 org.omg.CORBA.portable.Streamable  
{  
 // field that holds an Account object  
 public Account value = null;  
  
 // default constructor  
 public AccountHolder ()  
 {  
 }  
   
 // creates a new AccountHolder from initialValue  
 public AccountHolder (Account initialValue)  
 {  
 value = initialValue;  
 }  
   
 // reads the contents of i and assigns the contents to value  
 public void \_read (org.omg.CORBA.portable.InputStream i)  
 {  
 value = AccountHelper.read (i);  
 }  
  
 // writes value to o  
 public void \_write (org.omg.CORBA.portable.OutputStream o)  
 {  
 AccountHelper.write (o, value);  
 }  
   
 // returns the typecode for Account  
 public org.omg.CORBA.TypeCode \_type ()  
 {  
 return AccountHelper.type ();  
 }  
  
}

For more information on Holder classes, see Chapter 1.4, *Mapping for Basic Types* in the  [*OMG IDL to Java Language Mapping*](http://cgi.omg.org/cgi-bin/doc?ptc/00-01-08). The Holder classes defined in the package org.omg.CORBA are:

     AnyHolder  
     AnySeqHolder  
     BooleanHolder  
     BooleanSeqHolder  
     ByteHolder  
     CharHolder  
     CharSeqHolder  
     CurrentHolder  
     DoubleHolder  
     DoubleSeqHolder  
     FixedHolder  
     FloatHolder  
     FloatSeqHolder  
     IntHolder  
     LongHolder  
     LongLongSeqHolder  
     LongSeqHolder  
     ObjectHolder  
     OctetSeqHolder  
     ParameterModeHolder  
     PolicyErrorHolder  
     PolicyListHolder  
     PrincipalHolder  
     ServiceInformationHolder  
     ShortHolder  
     ShortSeqHolder  
     StringHolder  
     StringSeqHolder  
     TypeCodeHolder  
     ULongLongSeqHolder  
     ULongSeqHolder  
     UnknownUserExceptionHolder  
     UShortSeqHolder  
     ValueBaseHolder  
     WCharSeqHolder  
     WrongTransactionHolder  
     WStringSeqHolder

## Helper Classes

Helper files supply several static methods needed to manipulate the type. These include:

* Any insert and extract operations for the type
* getting the repository id
* getting the typecode
* reading and writing the type from and to a stream
* implement the ValueHelper interface (if it is a user-defined value type)

The helper class for a mapped IDL interface or abstract interface also include narrow operation(s). The static narrow method allows an org.omg.CORBA.Object to be narrowed to the object reference of a more specific type. The IDL exception CORBA.BAD\_PARAM is thrown if the narrow fails because the object reference does not support the requested type. A different system exception is raised to indicate other kinds of errors. Trying to narrow a null will always succeed with a return value of null. Generally, the only helper method an application programmer uses is the narrow method. The other methods are normally used behind the scenes and are transparent to the programmer.

Helper classes fall into two broad categories, [helpers for value types](#26in1rg) and [helpers for non value types](#tyjcwt). Because all of the helper classes in one category provide the same methods, one generic explanation of each category of helper classes is presented here.

When OMG IDL is mapped to the Java programming language, a "helper" class is generated for each user-defined type. This generated class will have the name of the user-defined type with the suffix Helper appended. For example, if the interface Account is defined in OMG IDL, the idlj compiler will automatically generate a class named AccountHelper. The AccountHelper class will contain the static methods needed for manipulating instances of the type, in this case, Account objects.

### The narrow Method

When an object is the return value for a method, it is returned in the form of a generic object, either an org.omg.CORBA.Object object or a java.lang.Object object. This object must be cast to its more specific type before it can be operated on. For example, an Account object will be returned as a generic object and must be narrowed to an Account object so that Account methods may be called on it.

The narrow method has two forms, one that takes an org.omg.CORBA.Object object and one that takes a java.lang.Object object. Whether the interface is abstract or not determines which narrow method its helper class will provide. The helper class for an interface that is not abstract will have a narrow method that takes a CORBA object, whereas the narrow method for an interface that is abstract will take an object in the Java programming language. The helper class for a non-abstract interface that has at least one abstract base interface will provide both versions of the narrow method.

The [*Hello World*](http://docs.google.com/technotes/guides/idl/jidlExample.html) tutorial uses a narrow method that looks like this:

// create and initialize the ORB  
 ORB orb = ORB.init(args, null);  
  
 // get the root naming context  
 org.omg.CORBA.Object objRef =   
 orb.resolve\_initial\_references("NameService");  
 // Use NamingContextExt instead of NamingContext. This is   
 // part of latest Inter-Operable naming Service.   
 NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);  
   
 // resolve the Object Reference in Naming  
 String name = "Hello";  
 helloImpl = HelloHelper.narrow(ncRef.resolve\_str(name));

### Example of a Basic Helper Class

A basic helper class, for purposes of this explanation, is one with the methods that are provided by every helper class, plus a narrow method if the type defined in OMG IDL maps to an interface in the Java programming language. Types that are not value types will have a basic helper class generated for them.

For example, assuming that the interface Account is not a value type IDL type and is also not an abstract interface and has no abstract base interfaces, its AccountHelper class will look like this:

abstract public class AccountHelper  
{  
 private static String \_id = "IDL:Account:1.0";  
  
 // inserts an Account object into an Any object  
 public static void insert (org.omg.CORBA.Any a, Account that)  
 {  
 org.omg.CORBA.portable.OutputStream out = a.create\_output\_stream ();  
 a.type (type ());  
 write (out, that);  
 a.read\_value (out.create\_input\_stream (), type ());  
 }  
  
 // extracts an Account object from an Any object  
 public static Account extract (org.omg.CORBA.Any a)  
 {  
 return read (a.create\_input\_stream ());  
 }  
  
   
 private static org.omg.CORBA.TypeCode \_\_typeCode = null;  
 // gets the typecode for this type  
 synchronized public static org.omg.CORBA.TypeCode type ()  
 {  
 if (\_\_typeCode == null)  
 {  
 \_\_typeCode = org.omg.CORBA.ORB.init ().create\_interface\_tc (AccountHelper.id (), "Account");  
 }  
 return \_\_typeCode;  
 }  
  
 // gets the repository id for this type  
 public static String id ()  
 {  
 return \_id;  
 }  
  
 // reads an Account object from an input stream  
 public static Account read (org.omg.CORBA.portable.InputStream istream)  
 {  
 return narrow (istream.read\_Object (\_AccountStub.class));  
 }  
  
 // writes an Account object to an outputstream  
 public static void write (org.omg.CORBA.portable.OutputStream ostream, Account value)  
 {  
 ostream.write\_Object ((org.omg.CORBA.Object) value);  
 }  
  
 // converts (narrows) an Object to an Account object  
 public static Account narrow (org.omg.CORBA.Object obj)  
 {  
 if (obj == null)  
 return null;  
 else if (obj instanceof Account)  
 return (Account)obj;  
 else if (!obj.\_is\_a (id ()))  
 throw new org.omg.CORBA.BAD\_PARAM ();  
 else  
 {  
 org.omg.CORBA.portable.Delegate delegate = ((org.omg.CORBA.portable.ObjectImpl)obj).\_get\_delegate ();  
 \_AccountStub stub = new \_AccountStub ();  
 stub.\_set\_delegate(delegate);  
 return stub;  
 }  
 }  
  
}

### Value Type Helper Classes

A helper class for a value type includes different renderings of the same methods generated for non-value type methods. The main difference is that value types are types that can be passed by value as parameters or return values of a method, which means that they must be serializable.

Assuming that Address is a value type, the AddressHelper class will look like this:

abstract public class AddressHelper  
{  
 private static String \_id = "IDL:Address:1.0";  
  
 // same as for non-value type  
 public static void insert (org.omg.CORBA.Any a, Address that)  
 {  
 org.omg.CORBA.portable.OutputStream out = a.create\_output\_stream ();  
 a.type (type ());  
 write (out, that);  
 a.read\_value (out.create\_input\_stream (), type ());  
 }  
  
 // same as for non-value type  
 public static Address extract (org.omg.CORBA.Any a)  
 {  
 return read (a.create\_input\_stream ());  
 }  
  
 private static org.omg.CORBA.TypeCode \_\_typeCode = null;  
 private static boolean \_\_active = false;  
   
 // getting the typecode for the type  
 synchronized public static org.omg.CORBA.TypeCode type ()  
 {  
 if (\_\_typeCode == null)  
 {  
 synchronized (org.omg.CORBA.TypeCode.class)  
 {  
 if (\_\_typeCode == null)  
 {  
 if (\_\_active)  
 {  
 return org.omg.CORBA.ORB.init().create\_recursive\_tc ( \_id );  
 }  
 \_\_active = true;  
 org.omg.CORBA.ValueMember[] \_members0 = new org.omg.CORBA.ValueMember[0];  
 org.omg.CORBA.TypeCode \_tcOf\_members0 = null;  
 \_\_typeCode = org.omg.CORBA.ORB.init ().create\_value\_tc (\_id, "Address", org.omg.CORBA.VM\_NONE.value, null, \_members0);  
 \_\_active = false;  
 }  
 }  
 }  
 return \_\_typeCode;  
 }  
  
 // same as for non-value type  
 public static String id ()  
 {  
 return \_id;  
 }  
  
 // reads a serializable instance of Address from the given input stream  
 public static Address read (org.omg.CORBA.portable.InputStream istream)  
 {  
 return (Address)((org.omg.CORBA\_2\_3.portable.InputStream) istream).read\_value (id ());  
 }  
  
 // writes a serializable instance of Address to the given output stream  
 public static void write (org.omg.CORBA.portable.OutputStream ostream, Address value)  
 {  
 ((org.omg.CORBA\_2\_3.portable.OutputStream) ostream).write\_value (value, id ());  
 }  
  
  
}

The Helper classes defined in the package org.omg.CORBA are:

     AnySeqHelper  
     BooleanSeqHelper  
     CharSeqHelper  
     CompletionStatusHelper  
     CurrentHelper  
     DefinitionKindHelper  
     DoubleSeqHelper  
     FieldNameHelper  
     FloatSeqHelper  
     IdentifierHelper  
     IDLTypeHelper  
     LongLongSeqHelper  
     LongSeqHelper  
     NameValuePairHelper  
     ObjectHelper  
     OctetSeqHelper  
     ParameterModeHelper  
     PolicyErrorCodeHelper  
     PolicyErrorHelper  
     PolicyHelper  
     PolicyListHelper  
     PolicyTypeHelper  
     RepositoryIdHelper  
     ServiceDetailHelper  
     ServiceInformationHelper  
     SetOverrideTypeHelper  
     ShortSeqHelper  
     StringSeqHelper  
     StringValueHelper  
     StructMemberHelper  
     ULongLongSeqHelper  
     ULongSeqHelper  
     UnionMemberHelper  
     UnknownUserExceptionHelper  
     UShortSeqHelper  
     ValueBaseHelper  
     ValueMemberHelper  
     VersionSpecHelper  
     VisibilityHelper  
     WCharSeqHelper  
     WrongTransactionHelper  
     WStringSeqHelper  
     WStringValueHelper

# Other Classes

The other classes and interfaces in the CORBA package, which are used behind the scenes, can be put into four groups. Three of the groups are used with requests in some capacity, and the fourth group, concerning the Interface Repository, is a category by itself.

## Classes Created by an ORB

The first group contains classes that are created by an ORB and contain information used in request operations.

* TCKind -- indicates the kind (datatype) for a TypeCode object
* TypeCode -- indicates a datatype and possibly other information
* Any -- contains a value and its typecode
* NamedValue -- contains a name, an Any object, and an argument mode flag. NamedValue objects contain information about method arguments, method return values, or a context.
* ContextList -- a list of strings that describe the contexts that need to be resolved and sent with an invocation
* ExceptionList -- a list of TypeCodes for exceptions that may be thrown by a method
* Environment -- a container for the exception thrown during a method invocation
* Context -- a list of NamedValue objects used to pass auxiliary information from client to server
* NVList -- a list of NamedValue objects, used to pass arguments or get results

## Classes That Deal with Requests

The second group of classes deals with requests:

* Object -- the base class for all CORBA object references
* Request -- the main class in the DII, which contains methods for adding arguments to the request, for accessing information about the method being invoked (the method name, its arguments, exceptions it throws, and so on), and for making invocations on the request
* DynamicImplementation -- the base class for server implementations using the DSI. It has the method invoke, which is used by an implementation of this class to determine the state of a ServerRequest object and to set its result or exception
* ServerRequest -- captures the explicit state of a request for the Dynamic Skeleton Interface

## Interfaces That Serve as Constants

The third group contains interfaces that serve as constants. The IDL-to-Java mapping mandates that IDL enums are mapped to a Java class with the enumerated values represented as public static final fields in that class (e.g. DefinitionKind). On the other hand IDL constants defined outside of an IDL interface are mapped to a Java interface for each constant.

This is why several interfaces in the org.omg.CORBA package consist of a single field, value, which is a short. This field is a constant used for such things as an error code or value modifier. For example, the value field of the interface BAD\_POLICY is one of the possible reasons for the exception PolicyError to be thrown. To specify this error code, you would use BAD\_POLICY.value.

The exception PolicyError uses the value field of the following interfaces as its possible error codes.

* BAD\_POLICY
* BAD\_POLICY\_TYPE
* BAD\_POLICY\_VALUE
* UNSUPPORTED\_POLICY
* UNSUPPORTED\_POLICY\_VALUE

The method TypeCode.type\_modifier returns the value field of one of the following interfaces. The VM in the names of these interfaces stands for "value modifier."

* VM\_NONE
* VM\_ABSTRACT
* VM\_CUSTOM
* VM\_TRUNCATABLE

The following constants are returned by a ValueMember object's access method to denote the visibility of the ValueMember object.

* PRIVATE\_MEMBER
* PUBLIC\_MEMBER

These flags, used in NamedValue objects or as parameters to methods, are defined in the following interfaces:

* ARG\_IN
* ARG\_INOUT
* ARG\_OUT
* CTX\_RESTRICT\_SCOPE

## Interface Repository Interfaces and Classes

A fourth group contains the Interface Repository interfaces and classes, which are generated by the idlj compiler from the OMG IDL interface ir.idl. The purpose of the Interface Repository is to identify the interfaces stored in it so that they can be accessed by an ORB. Each module, type, interface, attribute, operation, parameter, exception, constant, and so on is described completely by the Interface Repository API.

An ORB does not require that there be an interface repository, and Java IDL does not include one. Even though this release does not include an implementation of an interface repository, the following IR classes and interfaces have been included for the purpose of creating typecodes (see create\_value\_tc, create\_struct\_tc, create\_union\_tc and create\_exception\_tc methods in interface org.omg.CORBA.ORB):

&nbs

* IRObject
* IDLType
* DefinitionKind
* StructMember
* UnionMember
* ValueMember

# Related Documentation

For overviews, guides, and a tutorial, please see:

* [Java IDL home page](http://docs.google.com/technotes/guides/idl/index.html)

# CORBA Features Not Implemented in Java IDL

Some of the API included in org.omg subpackages is provided for conformance with the current OMG CORBA specification but is not implemented in Sun's release of the JDKTM. This enables other JDK licensees to provide implementations of this API in standard extensions and products.

## Features That Throw NO\_IMPLEMENT

Some of the API included in org.omg subpackages throw NO\_IMPLEMENT exceptions for various reasons. Among these reasons are:

* In some cases, for example LocalObject, the complete implementation according to the specification indicates that these API should throw NO\_IMPLEMENT.
* In most cases, for example methods in ORB.java, methods that throw NO\_IMPLEMENT are actually implemented in subclasses elsewhere in the ORB code.
* In some cases, for example \_get\_interface\_def() and \_get\_interface, API are really not yet implemented.

## General Summary of Features or API Not Implemented in This Release:

* Interface Repository. An Interface Repository is not required for normal operation of Java IDL.
* Java IDL does not support long double.
* Policies ([org.omg.CORBA.Policy](http://docs.google.com/Policy.html)) and methods for getting them are not implemented.
* Domain managers ([org.omg.CORBA.DomainManager](http://docs.google.com/DomainManager.html)) and methods for getting them are not implemented.
* Service Information [org.omg.CORBA.ServiceInformation](http://docs.google.com/ServiceInformation.html) and ORB method public boolean get\_service\_information(short service\_type, ServiceInformationHolder service\_info) are not implemented.
* ORB methods for supporting single-threading (perform\_work, work\_pending) are not implemented.
* IDL contexts.

## Specific List of Unimplemented Features in Package org.omg.CORBA

### Unimplemented Methods in package org.omg.CORBA:

* ORB
  + public org.omg.CORBA.Policy create\_policy(int type, org.omg.CORBA.Any val)
  + public void perform\_work()
  + public boolean work\_pending()
  + public org.omg.CORBA.Current get\_current()
  + create\_operation\_list
  + get\_default\_context
  + get\_service\_information
  + obsolete DynAnys (deprecated in favor of DynamicAny package)

**Since:** JDK1.2

| | [**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/ietf/jgss/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/org/omg/CORBA_2_3/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?org/omg/CORBA/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).