Ice Slice API

Ice

Overview

module Ice

The Ice core library. Among many other features, the Ice core library manages all the communication tasks using an efficient protocol (including protocol compression and support for both TCP and UDP), provides a thread pool for multi-threaded servers, and additional functionality that supports high scalability.

Class Index

ConnectionInfo — Base class providing access to the connection details.
EndpointInfo — Base class providing access to the endpoint details.
IPConnectionInfo — Provides access to the connection details of an IP connection
IPEndpointInfo — Provides access to the address details of a IP endpoint.
OpaqueEndpointInfo — Provides access to the details of an opaque endpoint.
TCPConnectionInfo — Provides access to the connection details of a TCP connection
TCPEndpointInfo — Provides access to a TCP endpoint information.
UDPConnectionInfo — Provides access to the connection details of a UDP connection
UDPEndpointInfo — Provides access to an UDP endpoint information.

Interface Index

Communicator — The central object in Ice.
Connection — The user-level interface to a connection.
Endpoint — The user-level interface to an endpoint.
ImplicitContext — An interface to associate implict contexts with communicators.
Locator — The Ice locator interface.
LocatorRegistry — The Ice locator registry interface.
Logger — The Ice message logger.
ObjectAdapter — The object adapter provides an up-call interface from the Ice run time to the implementation of Ice objects.
ObjectFactory — A factory for objects.
Plugin — A communicator plug-in.
PluginManager — Each communicator has a plug-in manager to administer the set of plug-ins.
Process — An administrative interface for process management.
Properties — A property set used to configure Ice and Ice applications.
PropertiesAdmin — The PropertiesAdmin interface provides remote access to the properties of a communicator.
Router — The Ice router interface.
ServantLocator — A servant locator is called by an object adapter to locate a servant that is not found in its active servant map.
Stats — An interface Ice uses to report statistics, such as how much data is sent or received.

Exception Index

AdapterAlreadyActiveException — This exception is raised if a server tries to set endpoints for an adapter that is already active.
AdapterNotFoundException — This exception is raised if an adapter cannot be found.
AlreadyRegisteredException — An attempt was made to register something more than once with the Ice run time.
BadMagicException — This exception indicates that a message did not start with the expected magic number ('I', 'c', 'e', 'P').
CloneNotImplementedException — An attempt was made to clone a class that does not support cloning.
CloseConnectionException — This exception indicates that the connection has been gracefully shut down by the server.
CloseTimeoutException — This exception indicates a connection closure timeout condition.
CollocationOptimizationException — This exception is raised if a feature is requested that is not supported with collocation optimization.
CommunicatorDestroyedException — This exception is raised if the Communicator has been destroyed.
CompressionException — This exception indicates a problem with compressing or uncompressing data.
ConnectFailedException — This exception indicates connection failures.
ConnectTimeoutException — This exception indicates a connection establishment timeout condition.
ConnectionLostException — This exception indicates a lost connection.
ConnectionNotValidatedException — This exception is raised if a message is received over a connection that is not yet validated.
ConnectionRefusedException — This exception indicates a connection failure for which the server host actively refuses a connection.
ConnectionTimeoutException — This exception indicates that a connection has been shut down because it has been idle for some time.
DNSException — This exception indicates a DNS problem.
DatagramLimitException — A datagram exceeds the configured size.
EncapsulationException — This exception indicates a malformed data encapsulation.
EndpointParseException — This exception is raised if there was an error while parsing an endpoint.
EndpointSelectionTypeParseException — This exception is raised if there was an error while parsing an endpoint selection type.
FacetNotExistException — This exception is raised if no facet with the given name exists, but at least one facet with the given identity exists.
FeatureNotSupportedException — This exception is raised if an unsupported feature is used.
FileException — This exception indicates file errors.
FixedProxyException — This exception indicates that an attempt has been made to change the connection properties of a fixed proxy.
ForcedCloseConnectionException — This exception is raised by an operation call if the application forcefully closes the connection Ice::Connection::close.
IdentityParseException — This exception is raised if there was an error while parsing a stringified identity.
IllegalIdentityException — This exception is raised if an illegal identity is encountered.
IllegalMessageSizeException — This exception indicates that a message size is less than the minimum required size.
InitializationException — This exception is raised when a failure occurs during initialization.
InvalidReplicaGroupIdException — This exception is raised if the replica group provided by the server is invalid.
MarshalException — This exception is raised for errors during marshaling or unmarshaling data.
MemoryLimitException — This exception is raised if a request size exceeds the limit specified by the `Ice.MessageSizeMax` property.
NoEndpointException — This exception is raised if no suitable endpoint is available.
NoObjectFactoryException — This exception is raised if no suitable object factory was found during unmarshaling of a Slice class instance.
NotRegisteredException — An attempt was made to find or deregister something that is not registered with the Ice run time or Ice locator.
ObjectAdapterDeactivatedException — This exception is raised if an attempt is made to use a deactivated ObjectAdapter.
ObjectAdapterIdInUseException — This exception is raised if an ObjectAdapter cannot be activated.
ObjectNotExistException — This exception is raised if an object does not exist on the server, that is, if no facets with the given identity exist.
ObjectNotFoundException — This exception is raised if an object cannot be found.
OperationNotExistException — This exception is raised if an operation for a given object does not exist on the server.
PluginInitializationException — This exception indicates that a failure occurred while initializing a plug-in.
ProtocolException — A generic exception base for all kinds of protocol error conditions.
ProxyParseException — This exception is raised if there was an error while parsing a stringified proxy.
ProxyUnmarshalException — This exception is raised if inconsistent data is received while unmarshaling a proxy.
RequestFailedException — This exception is raised if a request failed.
ResponseSentException — Indicates that the response to a request has already been sent; re-dispatching such a request is not possible.
SecurityException — This exception indicates a failure in a security subsystem, such as the IceSSL plug-in.
ServerNotFoundException — This exception is raised if a server cannot be found.
SocketException — This exception indicates socket errors.
StringConversionException — This exception is raised when a string conversion to or from UTF-8 fails during marshaling or unmarshaling.
SyscallException — This exception is raised if a system error occurred in the server or client process.
TimeoutException — This exception indicates a timeout condition.
TwowayOnlyException — The operation can only be invoked with a twoway request.
UnexpectedObjectException — This exception is raised if the type of an unmarshaled Slice class instance does not match its expected type.
UnknownException — This exception is raised if an operation call on a server raises an unknown exception.
UnknownLocalException — This exception is raised if an operation call on a server raises a local exception.
UnknownMessageException — This exception indicates that an unknown protocol message has been received.
UnknownReplyStatusException — This exception indicates that an unknown reply status has been received.
UnknownRequestIdException — This exception indicates that a response for an unknown request ID has been received.
UnknownUserException — An operation raised an incorrect user exception.
UnmarshalOutOfBoundsException — This exception is raised if an out-of-bounds condition occurs during unmarshaling.
UnsupportedEncodingException — This exception indicates an unsupported data encoding version.
UnsupportedProtocolException — This exception indicates an unsupported protocol version.
VersionMismatchException — This exception is raised if the Ice library version does not match the version in the Ice header files.

Structure Index

Current — Information about the current method invocation for servers.
Identity — The identity of an Ice object.

Sequence Index

Dictionary Index

Constant Index

Enumeration Index

Sequences

sequence<bool> BoolSeq

A sequence of bools.

sequence<byte> ByteSeq

A sequence of bytes.

Used By

- Ice::BadMagicException::badMagic
- Ice::OpaqueEndpointInfo::rawBytes
- IcePatch2::ByteSeqSeq
- IcePatch2::FileInfo::checksum
- IcePatch2::FileServer::getChecksum
- IcePatch2::FileServer::getFileCompressed

sequence<double> DoubleSeq

A sequence of doubles.

local sequence<Endpoint> EndpointSeq

A sequence of endpoints.

Used By

- Ice::ObjectAdapter::getEndpoints
- Ice::ObjectAdapter::getPublishedEndpoints

sequence<float> FloatSeq

A sequence of floats.

sequence<Identity> IdentitySeq

A sequence of identities.

Used By

- Glacier2::IdentitySet::add
- Glacier2::IdentitySet::get
- Glacier2::IdentitySet::remove

sequence<int> IntSeq

A sequence of ints.

sequence<long> LongSeq

A sequence of longs.

sequence<Object*> ObjectProxySeq

A sequence of object proxies.

Used By

- Ice::Router::addProxies
- IceGrid::Query::findAllObjectsByType
- IceGrid::Query::findAllReplicas

sequence<Object> ObjectSeq

A sequence of objects.

sequence<short> ShortSeq

A sequence of shorts.

sequence<string> StringSeq

A sequence of strings.

Used By

- Glacier2::SSLInfo::certs
- Glacier2::StringSet::add
- Glacier2::StringSet::get
- Glacier2::StringSet::remove
- Ice::PluginManager::getPlugins
- Ice::Properties::getCommandLineOptions
- Ice::Properties::getPropertyAsListWithDefault
- Ice::Properties::getPropertyAsList
- Ice::Properties::parseCommandLineOptions
- Ice::Properties::parseIceCommandLineOptions
- IceBox::Service::start
- IceBox::ServiceObserver::servicesStarted
- IceBox::ServiceObserver::servicesStopped
- IceGrid::Admin::getAllAdapterIds
- IceGrid::Admin::getAllApplicationNames
- IceGrid::Admin::getAllNodeNames
- IceGrid::Admin::getAllRegistryNames
- IceGrid::Admin::getAllServerIds
- IceGrid::ApplicationUpdateDescriptor::removeNodes
- IceGrid::ApplicationUpdateDescriptor::removePropertySets
- IceGrid::ApplicationUpdateDescriptor::removeReplicaGroups
- IceGrid::ApplicationUpdateDescriptor::removeServerTemplates
- IceGrid::ApplicationUpdateDescriptor::removeServiceTemplates
- IceGrid::ApplicationUpdateDescriptor::removeVariables
- IceGrid::CommunicatorDescriptor::logs

- IceGrid::DistributionDescriptor::directories
- IceGrid::FileIterator::read
- IceGrid::NodeUpdateDescriptor::removePropertySets
- IceGrid::NodeUpdateDescriptor::removeServers
- IceGrid::NodeUpdateDescriptor::removeVariables
- IceGrid::PatchException::reasons
- IceGrid::PropertySetDescriptor::references
- IceGrid::ServerDescriptor::envs
- IceGrid::ServerDescriptor::options
- IceGrid::TemplateDescriptor::parameters
- IceSSL::ConnectionInfo::certs

Dictionaries

dictionary<string, string> Context

A request context. `Context` is used to transmit metadata about a request from the server to the client, such as Quality-of-Service (QoS) parameters. Each operation on the client has a `Context` as its implicit final parameter.

Used By

- Ice::Current::ctx
- Ice::ImplicitContext::getContext
- Ice::ImplicitContext::setContext

local dictionary<string, Object> FacetMap

A mapping from facet name to servant.

Used By

- Ice::ObjectAdapter::findAllFacets
- Ice::ObjectAdapter::removeAllFacets

local dictionary<Identity, Object> ObjectDict

A mapping between identities and Ice objects.

dictionary<string, string> PropertyDict

A simple collection of properties, represented as a dictionary of key/value pairs. Both key and value are strings.

Used By

- Ice::Communicator::proxyToProperty
- Ice::Properties::getPropertiesForPrefix
- Ice::PropertiesAdmin::getPropertiesForPrefix

See Also

- Ice::Properties::getPropertiesForPrefix

dictionary<string, string> SliceChecksumDict

A mapping from type IDs to Slice checksums. The dictionary allows verification at run time that client and server use matching Slice definitions.

Used By

- IceBox::ServiceManager::getSliceChecksums
- IceGrid::Admin::getSliceChecksums
- IceStorm::TopicManager::getSliceChecksums

Constants

const short TCPEndpointType = 1;

Uniquely identifies TCP endpoints.

const short UDPEndpointType = 3;

Uniquely identifies UDP endpoints.
