Connection Closure

The Ice run time may close a connection for many reasons, including the situations listed below:

- When deactivating an object adapter or shutting down a communicator
- As required by active connection management
- When initiated by an application
- After a request times out
- In response to an exception, such as a socket failure or protocol error

In most cases, the Ice run time closes a connection gracefully as required by the Ice protocol. The Ice run time only closes a connection forcefully when a timeout occurs or when the application explicitly requests it.

On this page:

Graceful Connection Closure

Gracefully closing a connection occurs in stages:

- In the process that initiates closure, incoming and outgoing requests that are in progress are allowed to complete, and then a close connection message is sent to the peer. Any incoming requests received after closure is initiated are silently discarded (but may be retried, as discussed in the next bullet). An attempt to make a new outgoing request on the connection results in a `CloseConnectionException` and an automatic retry (if enabled).

- Upon receipt of a close connection message, the Ice run time in the peer closes its end of the connection. Any outgoing requests still pending on that connection fail with a `CloseConnectionException`. This exception indicates to the Ice run time that it is safe to retry those requests without violating at-most-once semantics, assuming automatic retries have not been disabled.

- After detecting that the peer has closed the connection, the initiating Ice run time closes the connection.

Connection Closure and Oneway Invocations

Oneway invocations are generally considered reliable because they are sent over a stream-oriented transport. However, it is quite possible for oneway requests to be silently discarded if a server has initiated graceful connection closure. Whereas graceful closure causes a discarded twoway request to receive a `CloseConnectionException` and eventually be retried, the sender receives no notice about a discarded oneway request.

If an application makes assumptions about the reliability of oneway requests, it may be necessary to control the events surrounding connection closure as much as possible, for example by disabling active connection management and avoiding explicit connection closures.

See Also

- Active Connection Management
- Using Connections
- Protocol Messages
- Connection Establishment
- Oneway Invocations
- Automatic Retries