

Ice-Instrumentation-CommunicatorObserver

Ice::Instrumentation::CommunicatorObserver

Overview

local interface CommunicatorObserver

The communicator observer interface used by the Ice run-time to obtain and update observers for its observable objects. This interface should be implemented by add-ins that wish to observe Ice objects in order to collect statistics. An instance of this interface can be provided to the Ice run-time through the Ice communicator initialization data.

Used By

- [Ice::Communicator::getObserver](#)

Operation Index

[getConnectionEstablishmentObserver](#) — This method should return an observer for the given endpoint information and connector.

[getEndpointLookupObserver](#) — This method should return an observer for the given endpoint information.

[getConnectionObserver](#) — This method should return a connection observer for the given connection.

[getThreadObserver](#) — This method should return a thread observer for the given thread.

[getInvocationObserver](#) — This method should return an invocation observer for the given invocation.

[getDispatchObserver](#) — This method should return a dispatch observer for the given dispatch.

[setObserverUpdater](#) — The Ice run-time calls this method when the communicator is initialized.

Operations

[Ice::Instrumentation::Observer](#) [getConnectionEstablishmentObserver](#)([Ice::Endpoint](#) endpt, string connector)

This method should return an observer for the given endpoint information and connector. The Ice run-time calls this method for each connection establishment attempt.

Parameters

endpt — The endpoint.

connector — The description of the connector. For IP transports, this is typically the IP address to connect to.

Return Value

The observer to instrument the connection establishment.

[Ice::Instrumentation::Observer](#) [getEndpointLookupObserver](#)([Ice::Endpoint](#) endpt)

This method should return an observer for the given endpoint information. The Ice run-time calls this method to resolve an endpoint and obtain the list of connectors. For IP endpoints, this typically involves doing a DNS lookup to obtain the IP addresses associated with the DNS name.

Parameters

endpt — The endpoint.

Return Value

The observer to instrument the endpoint lookup.

[Ice::Instrumentation::ConnectionObserver](#) [getConnectionObserver](#)([Ice::ConnectionInfo](#) c, [Ice::Endpoint](#) e, [Ice::Instrumentation::ConnectionState](#) s, [Ice::Instrumentation::ConnectionObserver](#) o)

This method should return a connection observer for the given connection. The Ice run-time calls this method for each new connection and for all the Ice communicator connections when `ObserverUpdater::updateConnections` is called.

Parameters

`c` — The connection information.
`e` — The connection endpoint.
`s` — The state of the connection.
`o` — The old connection observer if one is already set or a null reference otherwise.

Return Value

The connection observer to instrument the connection.

Ice::Instrumentation::ThreadObserver getThreadObserver(string parent, string id, Ice::Instrumentation::ThreadState s, Ice::Instrumentation::ThreadObserver o)

This method should return a thread observer for the given thread. The Ice run-time calls this method for each new thread and for all the Ice communicator threads when `ObserverUpdater::updateThreads` is called.

Parameters

`parent` — The parent of the thread.
`id` — The ID of the thread to observe.
`s` — The state of the thread.
`o` — The old thread observer if one is already set or a null reference otherwise.

Return Value

The thread observer to instrument the thread.

Ice::Instrumentation::InvocationObserver getInvocationObserver(Object* prx, string operation, Ice::Context ctx)

This method should return an invocation observer for the given invocation. The Ice run-time calls this method for each new invocation on a proxy.

Parameters

`prx` — The proxy used for the invocation.
`operation` — The name of the invocation.
`ctx` — The context specified by the user.

Return Value

The invocation observer to instrument the invocation.

Ice::Instrumentation::DispatchObserver getDispatchObserver(Ice::Current c, int size)

This method should return a dispatch observer for the given dispatch. The Ice run-time calls this method each time it receives an incoming invocation to be dispatched for an Ice object.

Parameters

`c` — The current object as provided to the Ice servant dispatching the invocation.
`size` — The size of the dispatch.

Return Value

The dispatch observer to instrument the dispatch.

void setObserverUpdater(Ice::Instrumentation::ObserverUpdater updater)

The Ice run-time calls this method when the communicator is initialized. The add-in implementing this interface can use this object to get the Ice run-time to re-obtain observers for observed objects.

Parameters

`updater` — The observer updater object.