

IceStorm-Topic

IceStorm::Topic

Overview

interface Topic

Publishers publish information on a particular topic. A topic logically represents a type.

See Also

- [IceStorm::TopicManager](#)

Operation Index

[getName](#) — Get the name of this topic.
[getPublisher](#) — Get a proxy to a publisher object for this topic.
[getNonReplicatedPublisher](#) — Get a non-replicated proxy to a publisher object for this topic.
[subscribe](#) — Subscribe with the given `qos` to this topic. (*Deprecated*)
[subscribeAndGetPublisher](#) — Subscribe with the given `qos` to this topic.
[unsubscribe](#) — Unsubscribe the given `subscriber`.
[link](#) — Create a link to the given topic.
[unlink](#) — Destroy the link from this topic to the given topic `linkTo`.
[getLinkInfoSeq](#) — Retrieve information on the current links.
[getSubscribers](#) — Retrieve the list of subscribers for this topic.
[destroy](#) — Destroy the topic.

Operations

["nonmutating"] string getName()

Get the name of this topic.

Return Value

The name of the topic.

See Also

- [IceStorm::TopicManager::create](#)

["nonmutating"] Object* getPublisher()

Get a proxy to a publisher object for this topic. To publish data to a topic, the publisher calls [getPublisher](#) and then casts to the topic type. An unchecked cast must be used on this proxy. If a replicated IceStorm deployment is used this call may return a replicated proxy.

Return Value

A proxy to publish data on this topic.

["nonmutating"] Object* getNonReplicatedPublisher()

Get a non-replicated proxy to a publisher object for this topic. To publish data to a topic, the publisher calls [getPublisher](#) and then casts to the topic type. An unchecked cast must be used on this proxy.

Return Value

A proxy to publish data on this topic.

void subscribe(IceStorm::QoS theQoS, Object* subscriber)

Subscribe with the given `qos` to this topic. If the given `subscriber` proxy has already been registered, it will be replaced. Note that this can cause a loss of events to the subscribed object.

This operation is deprecated as of version 3.2.

subscribe is deprecated, use subscribeAndGetPublisher instead

Parameters

`theQoS` — The quality of service parameters for this subscription.
`subscriber` — The subscriber's proxy.

Return Value

The per-subscriber publisher object.

See Also

- [unsubscribe](#)

Object* subscribeAndGetPublisher([IceStorm::QoS](#) theQoS, Object* subscriber) throws [IceStorm::AlreadySubscribed](#), [IceStorm::BadQoS](#)

Subscribe with the given `qos` to this topic. A per-subscriber publisher object is returned.

Parameters

`theQoS` — The quality of service parameters for this subscription.
`subscriber` — The subscriber's proxy.

Return Value

The per-subscriber publisher object.

Exceptions

[IceStorm::AlreadySubscribed](#) — Raised if the subscriber object is already subscribed.
[IceStorm::BadQoS](#) — Raised if the requested quality of service is unavailable or invalid.

See Also

- [unsubscribe](#)

void unsubscribe(Object* subscriber)

Unsubscribe the given `subscriber`.

Parameters

`subscriber` — The proxy of an existing subscriber.

See Also

- [subscribe](#)

void link([IceStorm::Topic](#)* linkTo, int cost) throws [IceStorm::LinkExists](#)

Create a link to the given topic. All events originating on this topic will also be sent to `linkTo`.

Parameters

`linkTo` — The topic to link to.
`cost` — The cost to the linked topic.

Exceptions

[IceStorm::LinkExists](#) — Raised if a link to the same topic already exists.

void unlink([IceStorm::Topic](#)* linkTo) throws [IceStorm::NoSuchLink](#)

Destroy the link from this topic to the given topic `linkTo`.

Parameters

`linkTo` — The topic to destroy the link to.

Exceptions

[`IceStorm::NoSuchLink`](#) — Raised if a link to the topic does not exist.

["nonmutating"] [`IceStorm::LinkInfoSeq`](#) `getLinkInfoSeq()`

Retrieve information on the current links.

Return Value

A sequence of `LinkInfo` objects.

["nonmutating"] [`Ice::IdentitySeq`](#) `getSubscribers()`

Retrieve the list of subscribers for this topic.

Return Value

The sequence of Ice identities for the subscriber objects.

`void destroy()`

Destroy the topic.
