

Ice-ImplicitContext

Ice::ImplicitContext

Overview

local interface ImplicitContext

An interface to associate implicit contexts with communicators. When you make a remote invocation without an explicit `Ice::Context` parameter, Ice uses the per-proxy `Ice::Context` (if any) combined with the `ImplicitContext` associated with the communicator.

Ice provides several implementations of `ImplicitContext`. The implementation used depends on the value of the `Ice.ImplicitContext` property.

`None` (default) — No implicit context at all.

`PerThread` — The implementation maintains a `Ice::Context` per thread.

`Shared` — The implementation maintains a single `Ice::Context` shared by all threads.

`ImplicitContext` also provides a number of operations to create, update or retrieve an entry in the underlying context without first retrieving a copy of the entire context. These operations correspond to a subset of the `java.util.Map` methods, with `java.lang.Object` replaced by `String` and `null` replaced by the empty-string.

Used By

- `Ice::Communicator::getImplicitContext`

Operation Index

`getContext` — Get a copy of the underlying context.

`setContext` — Set the underlying context.

`containsKey` — Check if this key has an associated value in the underlying context.

`get` — Get the value associated with the given key in the underlying context.

`put` — Create or update a key/value entry in the underlying context.

`remove` — Remove the entry for the given key in the underlying context.

Operations

Ice::Context getContext()

Get a copy of the underlying context.

Return Value

A copy of the underlying context.

void setContext(Ice::Context newContext)

Set the underlying context.

Parameters

`newContext` — The new context.

bool containsKey(string key)

Check if this key has an associated value in the underlying context.

Parameters

`key` — The key.

Return Value

True if the key has an associated value, False otherwise.

string get(string key)

Get the value associated with the given key in the underlying context. Returns an empty string if no value is associated with the key. [containsKey](#) allows you to distinguish between an empty-string value and no value at all.

Parameters

`key` — The key.

Return Value

The value associated with the key.

string put(string key, string value)

Create or update a key/value entry in the underlying context.

Parameters

`key` — The key.

`value` — The value.

Return Value

The previous value associated with the key, if any.

string remove(string key)

Remove the entry for the given key in the underlying context.

Parameters

`key` — The key.

Return Value

The value associated with the key, if any.
