

# Freeze-Connection

---

## Freeze::Connection

### Overview

#### local interface Connection

A connection to a database (database environment with Berkeley DB). If you want to use a connection concurrently in multiple threads, you need to serialize access to this connection.

#### Used By

- [Freeze::Transaction::getConnection](#)

### Operation Index

`beginTransaction` — Create a new transaction.

`currentTransaction` — Returns the transaction associated with this connection.

`removeMapIndex` — Remove an old unused Freeze Map index @throws `IndexNotFoundException` Raised if this index does not exist  
`close` — Closes this connection.

`getCommunicator` — Returns the communicator associated with this connection.

`getEncoding` — Returns the encoding version used to encode the data.

`getName` — The name of the connected system (for example, the Berkeley DB environment).

### Operations

#### [Freeze::Transaction beginTransaction\(\)](#)

Create a new transaction. Only one transaction at a time can be associated with a connection.

#### Return Value

The new transaction.

#### Exceptions

`Freeze::TransactionAlreadyInProgressException` — Raised if a transaction is already associated with this connection.

#### [Freeze::Transaction currentTransaction\(\)](#)

Returns the transaction associated with this connection.

#### Return Value

The current transaction if there is one, null otherwise.

#### **void removeMapIndex(string mapName, string indexName)**

Remove an old unused Freeze Map index

#### Exceptions

`Freeze::IndexNotFoundException` — Raised if this index does not exist

#### **void close()**

Closes this connection. If there is an associated transaction, it is rolled back.

#### [Ice::Communicator getCommunicator\(\)](#)

Returns the communicator associated with this connection.

**Ice::EncodingVersion getEncoding()**

Returns the encoding version used to encode the data.

**string getName()**

The name of the connected system (for example, the Berkeley DB environment).

---