

# Freeze Maps

A Freeze map is a persistent, associative container in which the key type must be a [legal dictionary key type](#) and the value type can be any primitive or user-defined Slice type. For each pair of key and value types, the developer uses a code-generation tool to produce a language-specific class that conforms to the standard conventions for maps in that language. For example, in C++, the generated class resembles a `std::map`, and in Java it implements the `java.util.SortedMap` interface. Most of the logic for storing and retrieving state to and from the database is implemented in a Freeze base class. The generated map classes derive from this base class, so they contain little code and therefore are efficient in terms of code size.

You can only store data types that are defined in Slice in a Freeze map. Types without a Slice definition (that is, arbitrary C++ or Java types) cannot be stored because a Freeze map reuses the Ice-generated marshaling code to create the persistent representation of the data in the database. This is especially important to remember when defining a [Slice class](#) whose instances will be stored in a Freeze map; only the "public" (Slice-defined) data members will be stored, not the private state members of any derived implementation class.

## Topics

- [Freeze Map Concepts](#)
- [Using a Freeze Map in C++](#)
- [Using a Freeze Map in Java](#)
- [Using a Freeze Map in the File System Server](#)

## See Also

- [Classes](#)