Freeze Catalogs

In each database environment, Freeze maintains an internal table that contains type information describing all the databases in the environment. This table is an instance of a Freeze map in which the key is a string representing the database name and the value is an instance of `Freeze::CatalogData`:

Slice

```
module Freeze {
    struct CatalogData {
        bool evictor;
        string key;
        string value;
    };
};
```

An entry describes an evictor database if the `evictor` member is true, in which case the `key` and `value` members are empty strings. An entry that describes a Freeze map sets `evictor` to false; the `key` and `value` members contain the Slice types used when the map was defined.

FreezeScript tools such as `transformdb` and `dumpdb` access the catalog to obtain type information when none is supplied by the user. You can also use `dumpdb` to display the catalog of a database environment.

Freeze applications may access the catalog in the same manner as any other Freeze map. For example, the following C++ code displays the contents of a catalog:

C++

```
#include <Freeze/Catalog.h>
...
string envName = ...;
Freeze::ConnectionPtr conn = Freeze::createConnection(communicator, envName);
Freeze::Catalog catalog(conn, Freeze::catalogName());
for (Freeze::Catalog::const_iterator p = catalog.begin();
    p != catalog.end(); ++p) {
    if (p->second.evictor)
        cout << p->first << ": evictor" << endl;
    else
        cout << p->first << ": map<" << p->second.key
             << ", " << p->second.value << ">" << endl;
}
conn->close();
```

The equivalent Java code is shown below:

Java

```
String envName = ...;
Freeze.Connection conn = Freeze.Util.createConnection(communicator, envName);
Freeze.Catalog catalog = new Freeze.Catalog(conn, Freeze.Util.catalogName(), true);
for (java.util.Map.Entry<String, Freeze.CatalogData> e :
     catalog.entrySet()) {
    String name = e.getKey();
    Freeze.CatalogData data = e.getValue();
    if (data.evictor)
        System.out.println(name + ": evictor");
    else
        System.out.println(name + ": map<" + data.key + ", " + data.value + ">");
}
conn.close();
```

See Also