Java Mapping for Structures

On this page:

Basic Java Mapping for Structures

A Slice structure maps to a Java class with the same name. For each Slice data member, the Java class contains a corresponding public data member. For example, here is our Employee structure once more:

Slice

```
struct Employee {
    long number;
    string firstName;
    string lastName;
};
```

The Slice-to-Java compiler generates the following definition for this structure:

Java

```
public final class Employee implements java.lang.Cloneable, java.io.Serializable {
    public long number;
    public String firstName;
    public String lastName;

    public Employee() {}

    public Employee(long number, String firstName, String lastName) {
        this.number = number;
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public boolean equals(java.lang.Object rhs) {
        // ...
    }
    public int hashCode() {
        // ...
    }

    public java.lang.Object clone()
        java.lang.Object o;
        try
        {
            o = super.clone();
        }
        catch(java.lang.CloneNotSupportedException ex)
        {
            assert false; // impossible
        }
        return o;
    }
}
```

For each data member in the Slice definition, the Java class contains a corresponding public data member of the same name. Note that you can optionally customize the mapping for data members to use getters and setters instead.

The `equals` member function compares two structures for equality. Note that the generated class also provides the usual `hashCode` and `clone` methods. (`clone` has the default behavior of making a shallow copy.)

Java Default Constructors for Structures

Structures have a default constructor that default-constructs each data member. This means members of primitive type are initialized to the equivalent of zero, and members of reference type are initialized to null. Note that applications must always explicitly initialize members of structure and enumerated types because the Ice run time does not accept null as a legal value for these types.

If you wish to ensure that data members of primitive and enumerated types are initialized to specific values, you can declare default values in your Slice definition. The default constructor initializes each of these data members to its declared value.

Structures also have a second constructor that has one parameter for each data member. This allows you to construct and initialize a class instance in a single statement (instead of first having to construct the instance and then assign to its members).

See Also

- Structures
- Java Mapping for Enumerations
- Java Mapping for Sequences
- Java Mapping for Dictionaries
- Customizing the Java Mapping