Simple Classes

A Slice class definition is similar to a structure definition, but uses the class keyword. For example:

Apart from the keyword class, this definition is identical to the structure example. You can use a Slice class wherever you can use a Slice structure (but, as we will see shortly, for performance reasons, you should not use a class where a structure is sufficient). Unlike structures, classes can be empty:

```
Slice

class EmptyClass {}; // OK
struct EmptyStruct {}; // Error
```

Much the same design considerations as for empty interfaces apply to empty classes: you should at least stop and rethink your approach before committing yourself to an empty class.

You can specify a default value for a class data member that has one of the following types:

- An integral type (byte, short, int, long)
- A floating point type (float or double)
- string
- bool
- enum

For example:

```
Slice

class Location {
    string name;
    Point pt;
    bool display = true;
    string source = "GPS";
};
```

The legal syntax for literal values is the same as for Slice constants, and you may also use a constant as a default value. The language mapping guarantees that data members are initialized to their declared default values using a language-specific mechanism.

See Also

- Structures
- Constants and Literals