

Objective-C Mapping for Optional Data Members

The Objective-C mapping for [optional data members](#) in Slice [classes](#) and [exceptions](#) adds an extra boolean instance variable as well as two selectors for testing if the optional is set and clearing its value. The argument for the optional data member in the convenience constructor is mapped according to the [optional parameter mapping](#) for Slice operations. Consider the following Slice definition:

Slice

```
class C
{
    string name;
    optional(2) string alternateName;
    optional(5) bool active;
};
```

The generated Objective-C code provides the following API:

Objective-C

```
@interface testC : ICEObject
{
    NSString *name;
    NSString *alternateName;
    BOOL has_alternateName__;
    BOOL active;
    BOOL has_active__;
}
@property(n nonatomic, ICE_STRONG_ATTR) NSString *name;
@property(n nonatomic, ICE_STRONG_ATTR) NSString *alternateName;
@property(n nonatomic, assign) BOOL active;
-(id) init:(NSString*)name alternateName:(id)alternateName active:(id)active;
+(id) c:(NSString*)name alternateName:(id)alternateName active:(id)active;
+(id) c;
-(void)setAlternateName:(NSString*)alternateName_p;
-(BOOL)hasAlternateName;
-(void)clearAlternateName;
-(void)setActive:(BOOL)active_p;
-(BOOL)hasActive;
-(void)clearActive;
@end
```

Note that calling a get method or accessing the property when the value is not currently set returns an undefined value.

See Also

- [Optional Data Members](#)