Raising Exceptions in Objective-C

To throw an exception from an operation implementation, you simply allocate the exception, initialize it, and throw it. For example:

Objective-C

```
-(void) write:(NSMutableArray *)text current:(ICECurrent *)current
{
    // Try to write the file contents here...
    // Assume we are out of space...
    if (error)
        @throw [FSGenericError genericError:@"file too large"];
}
```

As for out-parameters and return values, you must take care to throw an autoreleased exception.

If you throw an arbitrary Objective-C exception that does not derive from `ICEException`, the client receives an `UnknownException`. Similarly, if you throw an "impossible" user exception (a user exception that is not listed in the exception specification of the operation), the client receives an `UnknownUserException`.

If you throw a run-time exception, such as `MemoryLimitException`, the client receives an `UnknownLocalException`. For that reason, you should never throw system exceptions from operation implementations. If you do, all the client will see is an `UnknownLocalException`, which does not tell the client anything useful.

> Three run-time exceptions are treated specially and not changed to `UnknownLocalException` when returned to the client: `ObjectNotExistException`, `OperationNotExistException`, and `FacetNotExistException`.

See Also

- Run-Time Exceptions
- Objective-C Mapping for Exceptions
- Server-Side Objective-C Mapping for Interfaces
- Parameter Passing in Objective-C