

Logging in iOS Applications

[fork and edit tutorial \(https://github.ibm.com/MFPSamples/DevCenter/tree/master/tutorials/en/foundation/8.0/analytics/remote-controlled-client-side-log-collection/ios.md\)](https://github.ibm.com/MFPSamples/DevCenter/tree/master/tutorials/en/foundation/8.0/analytics/remote-controlled-client-side-log-collection/ios.md) | [report issue \(https://github.ibm.com/MFPSamples/DevCenter/issues/new\)](https://github.ibm.com/MFPSamples/DevCenter/issues/new)

Overview

This tutorial provides the required code snippets in order to add logging capabilities in iOS applications.

Logging example

Outputs to Xcode console

```
#import "OCLogger.h"
+ (int) sum:(int) a with:(int) b{
    int sum = a + b;
    OCLogger* mathLogger = [OCLogger getInstanceWithPackage:@"MathUtils"];
    NSString* logMessage = [NSString stringWithFormat:@"sum called with args %d and %d. Returning %d",
a, b, sum];
    [mathLogger debug:logMessage];
    return sum;
}
```

Additional API Methods For Specific Tasks

Log capture is enabled by default. To turn log capture on or off:

```
OCLogger.setCapture(NO)
```

The default capture level is DEBUG in development and FATAL in production. To control the capture level (verbosity):

```
OCLogger.setLevel(OCLogger_DEBUG)
```

Log sending is enabled by default. To turn automatic log sending on or off:

```
OCLogger.setAutoSendLogs(NO)
```

For more information about the `Logger` API, see the API reference in the user documentation.