

Mobilogics

MobilogicsCore 5.0

Copyright 2009 - 2013 MobiLogics. All rights reserved.

Mon Oct 21 2013 15:00:20

Chapter 1

MobilogicsCore Release Notes

Mobilogics SDK Release Notes

Version 4.0 (2012-10-19)

Version 3.2 (2012-07-12)

- Support aScan.
- Support multi-accessory.
- Support iOS 6 beta 2 and later.
- Support Xcode 4.4 on OS X Lion.
- Support Xcode 4.5 beta on OS X Mountain Lion.
- Support ARC.
- Still support both Armv6 and Armv7 platform from iOS 4.x ~ 5.x
- Add Full SDK Manual within installed package.
- Remove Project Templates.

Version 2.4 (2012-04-25)

- Fixed compatible with Apple MessageUI framework and change Connection to MLConnection.
- Fixed some minor issue on BarcodeExample.
- Changed release package mode.

Version 2.3 (2012-03-16)

- Fixed some minor bug on iOS5.x sleep mode.
- Support both Armv6 and Armv7 platform from iOS 4.x ~ 5.x.

Version 2.2 (2011-12-22)

- Fixed Barcode Example source code can't work on iOS 5.0.1.
- Fixed iScan on iPad2 iOS 5.x will return empty string

Version 2.1 (2011-10-07)

- Fixed Barcode Framework CPU & Memory extra loading bug.

Version 2.0.2 (2011-10-05)

- Fixed BarcodeExample source code can't running bug.

Version 2.0.1 (2011-07-08)

- Fixed Connection object's behavior to balance cpu loading.
- Clone Connection class to MBConnection class to avoid some other framework had the same name class issue.

Version 2.0 (2011-05-09)

- Fixed Connection connectedAccessory method bug.
- Support Xcode 4.x (include Xcode 4.1 preview 4).

Version 1.5 (2011-04-18)

- Fixed iPad/iPhone/iPod connection compatible bug.
- Add more compatible with iOS 4.3.x.

Version 1.4 (2011-03-31)

- Add Accessory connection sensitive.

Version 1.3 (2011-03-17)

- Reduce Barcode Framework.
- Restructure Barcode Framework.
- Add External Accessory to DevTool Framework.

Version 1.2 (2011-02-19)

- Fixed minor bugs.

Version 1.1 (2011-02-18)

- SDK Lite add Barcode Example.
- Fixed SDK install scripts error.
- Support iOS v4.3 beta.

Version 1.0 (2011-02-16)

- Fixed minor bugs.

Version 1.0RC2 (2011-01-21)

- Add Barcode Example template.
- Fixed .pch header import setting.

Version 1.0RC1 (2011-01-21)

- Replace NSLog with LogError, LogWarn, LogInfo, LogVerbose functions.
- Fixed Navigation-based template fit new functions.
- Fixed DevTool framework header file to fit template.
- Fixed Barcode framework header file to fit template.
- Add OpenGL ES Application template support.
- Add Split View-based template support.
- Add Tab Bar Application template support.
- Add Utility Application template support.
- Add View-based Application template support.
- Add Window-based Application template support.

Version 0.9a.4 (2011-01-07)

- Fixed minor bugs.

Version 0.9a.3 (2011-01-07)

- Fixed frameowrk to support arm6 platform.

Version 0.9a.2 (2011-01-04)

- Fixed minor bugs.

Version 0.9a.1 (2011-01-03)

- Fixed Navigation-based Application template about within build Unit Test error.

Version 0.9a (2010-12-30)

- Add GHUnit framework to replace GTM Unit Test framework.
- Barcode framework change to new framework release type.
- DevTool framework change to new framework release type.
- Change Navigation-based Application template to fit all frameworks.

Version 0.9 (2010-12-28)

- Fixed Unit Test in iOS 4.2.1 compile error. After this version developer should compile in simulator mode before compile in device mode, because Unit Test can't show assert error in compile in device mode.
- Add MLog to replace NSLog.
- Fixed Navigation-based Application template error.
- Some other template in this version will not work completely, will fixed it next release.

Chapter 2

License

Mobilogics

Mobilogics SDK SOFTWARE LICENSE AGREEMENT

PLEASE READ THIS SOFTWARE LICENSE AGREEMENT ("LICENSE") CAREFULLY BEFORE USING Mobilogics SDK. BY USING Mobilogics SDK AS APPLICABLE, YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS LICENSE. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, DO NOT USE THE Mobilogics SDK.

Intellectual Property

As between you and Mobilogics, you agree and acknowledge that Mobilogics owns all rights, title and interest in Mobilogics SDK, including without limitation all associated Intellectual Property Rights. "Intellectual Property Rights" means any and all rights existing from time to time under patent law, copyright law, trade secret law, trademark law, unfair competition law, and any and all other proprietary rights, and any and all applications, renewals, extensions and restorations thereof, now or hereafter in force and effect worldwide. You agree to not remove, obscure, or alter Mobilogics or any third party's copyright notice, trademarks, or other proprietary rights notices affixed to or contained within or accessed in conjunction with or through Mobilogics SDK.

You agree that the Mobilogics SDK contain proprietary content, information and material that is owned by Mobilogics and/or its licensors, and is protected by applicable intellectual property and other laws, including but not limited to copyright. You agree that you will not use such proprietary content, information or materials other than for permitted use of Mobilogics SDK or in any manner that is inconsistent with the terms of this License or that infringes any intellectual property rights of a third party or Mobilogics. No portion of the Mobilogics SDK may be reproduced in any form or by any means. You agree not to modify, rent, lease, loan, sell, distribute, or create derivative works based on Mobilogics SDK, in any manner, and you shall not exploit Mobilogics SDK in any unauthorized way whatsoever, including but not limited to, using Mobilogics SDK to transmit any computer viruses, worms, trojan horses or other malware, or by trespass or burdening network capacity. You further agree not to use Mobilogics SDK in any manner to harass, abuse, stalk, threaten, defame or otherwise infringe or violate the rights of any other party, and that Mobilogics is not in any way responsible for any such use by you, nor for any harassing, threatening, defamatory, offensive, infringing or illegal messages or transmissions that you may receive as a result of using any of Mobilogics SDK.

Limitation of Liability.

TO THE EXTENT NOT PROHIBITED BY APPLICABLE LAW, IN NO EVENT SHALL Mobilogics BE LIABLE FOR PERSONAL INJURY, OR ANY INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES WHATSOEVER, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, CORRUPTION OR LOSS OF DATA, FAILURE TO TRANSMIT OR RECEIVE ANY DATA, BUSINESS INTERRUPTION OR ANY OTHER COMMERCIAL DAMAGES OR LOSSES, ARISING OUT OF OR RELATED TO YOUR USE OF OR INABILITY TO USE THE Mobilogics SDK AND SERVICES OR ANY THIRD PARTY SOFTWARE OR APPLICATIONS IN CONJUNCTION WITH THE Mobilogics SDK, HOWEVER CAUSED, REGARDLESS OF THE THEORY OF LIABILITY (CONTRACT, TORT OR OTHERWISE) AND EVEN IF Mobilogics HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS DO NOT ALLOW THE LIMITATION OF LIABILITY FOR PERSONAL

INJURY, OR OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION MAY NOT APPLY TO YOU.

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

NSObject(MLLogger)	11
--	----

Chapter 4

Class Documentation

4.1 NSObject(MLogger) Category Reference

Instance Methods

- (MLogger *) - [logger](#)

Class Methods

- (MLogger *) + [logger](#)

4.1.1 Detailed Description

Convenience methods for all NSObject classes. This category provides methods to obtain an MLogger instance from all classes and instances. You may want to override these methods in your local base class and provide caching local iVar, since these methods result in an NSDictionary lookup each time they are called. Actually it's not a bad hit, but in a high volume logging environment, it might make a difference.

Here is an example of what that might look like:

CODE TO ADD TO YOUR BASE CLASS .h file declarations

```
* MLogger *myLoggerIVar; // instance variable
*
```

CODE TO ADD TO YOUR BASE CLASS .m file

```
* static MLogger *myLoggerClassVar; // static "class" variable
*
* + (MLogger *) log
* {
*   if( myLoggerClassVar == nil ) {
*     myLoggerClassVar = [super mLogger];
*   }
*   return myLoggerClassVar;
* }
*
* - (MLogger *) log
* {
*   if( myLoggerIVar == nil ) {
*     myLoggerIVar = [super mLogger];
*   }
*   vreturn myLoggerIVar;
* }
*
```

For copyright & license, see COPYRIGHT.txt.

4.1.2 Method Documentation

+ (MLogger *) [logger](#)

Accessor for the MLogger instance to be used from within class methods.

- (MLLogger *) logger

Accessor for the MLLogger instance to be used from within instance methods. The default implementation returns the MLLogger for the class.

Index

logger

 NSObject(MLLogger), [11](#)

NSObject(MLLogger), [11](#)

 logger, [11](#)