FAQ and TroubleShooting iOS OData Client - Part 1

This document mostly targets the iOS client platform with subtle references to the SUP-ODP server. This applies to all the version of the SUP-ODP iOS client unless specified otherwise in the document. This document is divided into multiple parts. New sets of questions will be added as and when needed as separate documents.

Pramod.M
Lead iOS Developer
**SAP** Labs India Pvt.Ltd.

TABLE OF CONTENTS

INTRODUCTION

This document addresses the frequently asked questions on how to troubleshoot applications on the iOS client. Like already mentioned, most of this document is aimed at the iOS client but may, in some cases, refer to server troubleshooting topics as well. This topic will be covered in parts so that the document is concise and does not span too many topics.

TROUBLESHOOTING AND FREQUENTLY ASKED QUESTIONS

1. **I get an exception '*unrecognized selector sent to a class/instance*' when parsing the OData document.**

The SDMParser on iOS needs 2 linker flags to be set without which exceptions like the one mentioned above are bound to crop up. This can cause the application to crash. The solution for this is to include the 2 flags **-ObjC** and **–all_load** in the 'Other Linker Flags' option of the 'Build Settings' section in the Xcode target.
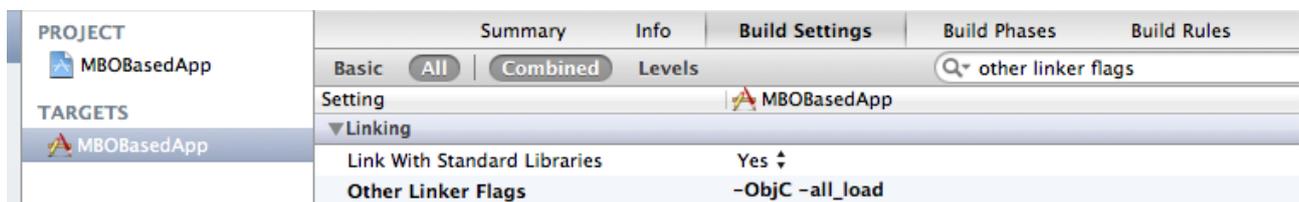
2. **I see errors (100s in number) during compilation of the iOS OData project.**

The SUP client libraries are dependent on some of the standard iOS frameworks and dylibs which need to be added to the project under the 'Build Phases' tab of the XCode target. For the full list of frameworks and libraries the needed to be added, please refer the documentation and the Sybase info center.
Please find the link for latest version here. Please refer to the links specific to the version for other version in the infocenter.
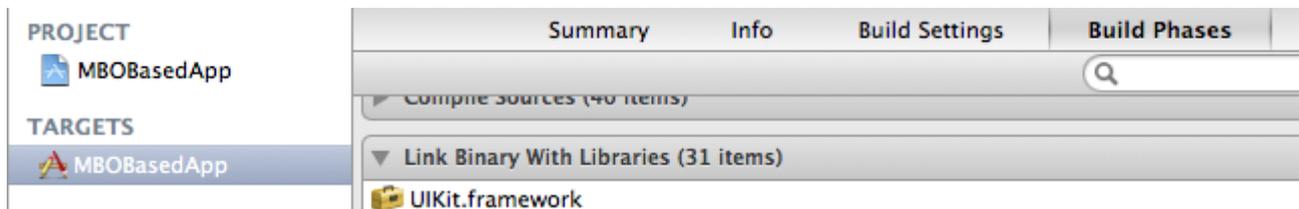
3. **I just upgraded my client project to v2.1.3 and I face compilation issues related to 'DataVault'**

With SUP OData SDK v2.1.3, the library libMO.a was split into libMO.a and libDataVault.a for reusability. Hence, when the application developer is upgrading is client project to v2.1.3, he needs to explicitly add the library 'libDataVault.a' into his project to avoid these compilation issues.

4. **When I build my application including all the client libraries I get 'Semantic Issue' for redefinition of enums.**

This can be caused because of importing both 'SUPRequest.h' and 'SDMHttpRequest.h' files within your project. To avoid this scenario a header with a common protocol 'SDMRequesting.h' has been provided. You need to include only this header file and instantiate the desired object using the **SDMRequestBuilder** class.

5. **While running the project on the simulator, I get a mail button in the SDMLogger logViewer screen, but it vanishes when I run it on a device.**

For the button to appear on your log-viewing screen, a mail account has to be mandatorily configured on your device, without which the libraries will not present this option to the application user. Once a valid mail account has been configured, it should continue to work like it did with the simulator.

6. **While using 'HTTPRequestType' as my request type, when I try to connect to a different back-end after making a request, it reuses the old credentials.**

When your request type is 'HTTPRequestType', the framework stores your HTTP session and persists your cookie by default. If the developer chooses to go against this, he needs to add 2 pieces of additional code to over-ride it.

```
[SDMRequestBuilder setRequestType:HTTPRequestType];
id<SDMRequesting> request = [SDMRequestBuilder requestWithURL:[NSURL URLWithString:@"URL"]];
[request setUseCookiePersistence:NO];
[request setUseSessionPersistence:NO];
```

7. **My request returns with an error response and I see in logs the error '*71002:*** -[__NSArrayM objectAtIndex:]: index 0 beyond bounds for empty array*'**

One of the main reasons this could be is because the URL that is being accessed might not be whitelisted within the SUP server. There is a high chance that you can connect to multiple back-end systems from the sample application. In this case all the URLs have to be whitelisted in the following section without which you will receive the error mentioned in the question.
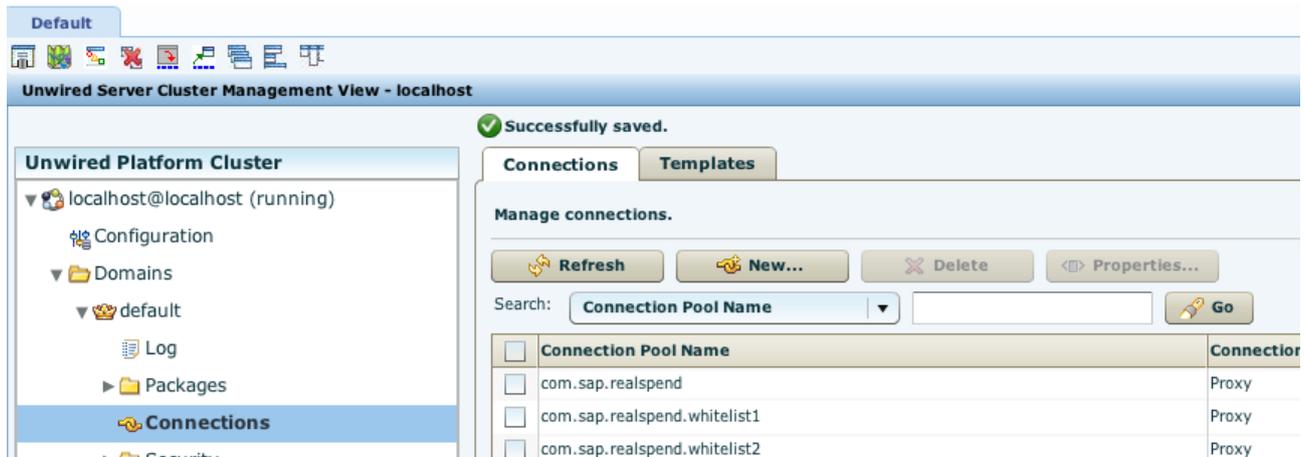
8. **I am trying to work with SDMDownloadCache with SUPRequestType and it is not working**

SUPRequest does not support caching as of versions 2.1.x and 2.2.x. Hence SDMDownloadCache will not work with SUPRequest in 2.1.x and ODPRequest in 2.2.x

9. **The SUP libraries are only giving error codes in some cases. How do I decipher the meaning of the code**

For error codes and their meaning, you need to either refer to the header file of LiteSUPUserManager class in 2.1.x and ODPUserManager class in 2.2.x or their corresponding header files.

10. **How do I get access to the client logs for trouble-shooting.**

The SUP OData SDK stores its log files in the 'Documents' folder under each of the application file system. In the simulator, you can access it under

/Users/<your user>/Library/Application Support/iPhone Simulator/<your version>/Applications/<you app>/Documents/OBData/mocaLog
AND
/Users/<your user>/Library/Application Support/iPhone Simulator/<your version>/Applications/<you app>/Documents/ODPLogs

The mocaLog gives the logs at the messaging level of the client and this is an important piece of information if you are reporting an issue to SAP.

The second set of logs is the client application level logs. You can provide these logs if you cannot access the console logs since principally both of them get written with the same logs.

In the device, you can always get the logs as soon as you connect your device and access the files using iTunes. For this you should allow sharing of files on iTunes in your XCode project configurations.

'**Application supports iTunes file sharing**' to YES.

www.sap.com