OMTP PUBLISHED

BONDI TESTCASE COVERAGE

VERSION:	Version 1.11
STATUS:	Approved Release
DATE OF LAST EDIT:	26th May 2010
OWNER:	OMTP Limited

The information contained in this document represents the current view held by OMTP Ltd. on the issues discussed as of the date of publication.

This document is provided “as is” with no warranties whatsoever including any warranty of merchantability, non-infringement, or fitness for any particular purpose. All liability (including liability for infringement of any property rights) relating to the use of information in this document is disclaimed. No license, express or implied, to any intellectual property rights are granted herein.

This document is distributed for informational purposes only and is subject to change without notice. Readers should not design products based solely on this document.

© 2010 OMTP Ltd. All rights reserved. OMTP and OMTP BONDI are registered trademarks of OMTP Ltd.

CONTENTS

1 INTRODUCTION 4
1.1. OBJECTIVES 4
2 THE REFERENCE IMPLEMENTATION 4
3 THE TEST SUITE 5
4 TEST COVERAGE 5
5 RESULTS 5
6 REFERENCES 9

1 INTRODUCTION

This document describes the state of the BONDI test cases in the context of the available reference implementation at release 1.11. This is expected to be the final release of the BONDI specifications.

1.1. OBJECTIVES

The primary objective of the work associated with test case coverage is to ensure that BONDI tests adequately exercise implementations. The test framework and the test suite provide the mechanism for running tests that verify correct operation of BONDI implementations.

For the purposes of this particular document, the aim was to ensure that the BONDI test suite provided adequate coverage for BONDI Release 1.11, the final release of the specification.

2 THE REFERENCE IMPLEMENTATION

The BONDI reference implementation is based on Windows Mobile. Not all aspects of BONDI 1.11 have been fully implemented in the implementation. In a number of cases, the reasons for the lack of implementation can be traced back to limitations in the underlying Windows Mobile platform.

Regardless of the reasons for the limitations, they provide some challenges in verifying the correctness of the test suite. Where verification of the tests could not be completed because of such limitations, it was carried out by manual inspection. This is indicated in the results presented.

The specific issues with the Windows Mobile-based implementation include:

The inability to access constants unless the objects that contain them are actually instantiated

The inability for errors to be thrown correctly in certain circumstances

The filesystem module implementation has not been upgraded to match the 1.11 specification, remaining mostly at 1.0 level. In addition, the implementation is incomplete

Filters have not been fully implemented in the telephony module, limiting the fields on which searches can be carried out

In the messaging module the following issues are known:

- Subscribing to and unsubscribing from binary SMS is not implemented
- Filter support is missing from searches of SMS, MMS and e-mails
- sendSMS sends an SMS but stores it in the wrong place

- subscriptions require additional filter support

In most cases, the remaining issues with the BONDI Windows Mobile-based reference implementation are unlikely to be corrected. The issues require access to personnel and skills no longer available to the BONDI work.

3 THE TEST SUITE

The test suite is accessed from the test page [1]. This page contains links to the latest specifications for BONDI, and to the pages and widgets that implement the tests. All tests were carried out using the pages, rather than the widgets.

Each module in BONDI has its own set of tests. These are exercised on a Windows Mobile handset on which the reference implementation had been installed. In most cases an HTC HD2 handset was used.

As the tests run, code in the test framework identifies the functions that have been invoked, from all of those available in the module. A record of the functions exercised is made, which can be uploaded once the test is complete. These records are available on-line [2].

4 TEST COVERAGE

As the tests run, code in the test framework identifies the functions that have been invoked, from all of those available in the module. A record of the functions exercised is made, which can be uploaded once the test is complete. These records are available on-line [2].

Because of the difficulties with the Windows Mobile-based implementation, discussed in *The Reference Implementation*, on page 4, the results of the coverage tests needed to be interpreted. In addition, to prevent these issues from affecting the validity of the coverage reports, some modifications were made to the tests during coverage testing. These temporary modifications will be reversed once coverage testing is complete

The modifications allowed some tests to run to completion which otherwise failed, preventing other, valid tests from being completed.

5 RESULTS

The results of coverage testing are presented in the tables below. Each table shows the BONDI modules and their components. Against each component is the estimated coverage.

Coverage is estimated automatically, where the Windows Mobile-based RI is able to execute the tests. Coverage reports are created during testing. The dates in the tables indicate the date of the coverage report on which the coverage estimate is based.

Coverage is reviewed manually where the Windows Mobile-based RI is unable to execute the tests. Coverage values that are estimated manually are shown in italics. The accompanying comments indicate the reason why the coverage needed to be estimated manually.

The current tables reflect the situation at the close of the BONDI testfest, held in Staines from May 17th -19th 2010.

In summary, the code coverage of the BONDI 1.11 tests is about 77%. Many of the modules have 100% coverage. The messaging module is still showing rather low coverage values. However, this is due to issues with the underlying BONDI implementation on Windows Mobile, rather than the test suite. We anticipate being able to verify much improved coverage values for these modules soon.

Module		Coverage	Comments
appconfig			
	AppConfigManager	100.0%	
applauncher			
	AppLauncherManager	100.0%	
bondi			
	bondi	100.0%	
calendar			
	CalendarManager	100.0%	
	Calendar	77.8%	
	Event	100.0%	
	EventProperties	100.0%	
camera			
	CameraError	*100.0%*	*RI/WebVM/WinMo issues prevent coverage tests reporting*
	CameraManager	100.0%	
	CaptureOptions	*100.0%*	*RI/WebVM/WinMo issues prevent coverage tests reporting*
	FeatureValue	*100.0%*	*RI/WebVM/WinMo issues prevent coverage tests reporting*
	CameraFeature	*100.0%*	*RI/WebVM/WinMo issues prevent coverage tests reporting*
	Camera	100.0%	
contact			
	ContactManager	100.0%	
	AddressBook	88.9%	
	Contact	0.0%	
	ContactProperties	83.3%	
	Address	0.0%	
	PhoneNumber	50.0%	
	EmailAddress	0.0%	
	ContactAddress	0.0%	
devicestatus			
	DeviceStatusManager	87.5%	
	AspectName	*100.0%*	*RI/WebVM/WinMo issues prevent coverage tests reporting*
	PropertyRef	*100.0%*	*RI/WebVM/WinMo issues prevent coverage tests reporting*

filesystem			
	FileSystemManager	100.0%	
	FileSystemListener	*100.0%*	*Incomplete implementation prevents coverage tests reporting correctly. Tests believed to be correct.*
	File	*100.0%*	*Incomplete implementation prevents coverage tests reporting correctly. Tests believed to be correct.*
	FileStream	*100.0%*	*Incomplete implementation prevents coverage tests reporting correctly. Tests believed to be correct.*
gallery			
	GalleryError	*100.0%*	*RI/WebVM/WinMo issues prevent coverage tests reporting correctly*
	GalleryManager	100.0%	
	Gallery	94.4%	
	MediaItem	56.0%	
geolocation			
	Geolocation	100.0%	
	Coordinates	*100.0%*	*RI/WebVM/WinMo issues prevent coverage tests reporting correctly*
	Position	*100.0%*	*RI/WebVM/WinMo issues prevent coverage tests reporting correctly*
	PositionError	*100.0%*	*RI/WebVM/WinMo issues prevent coverage tests reporting correctly*
messaging			
	MessagingError	*100.0%*	*RI/WebVM/WinMo issues prevent coverage tests reporting correctly*
	MessagingManager	30.3%	
	SMS	0.0%	
	MMSSlide	0.0%	
	MMS	0.0%	
	Email	100.0%	
	BinarySMS	0.0%	
task			
	TaskManager	100.0%	
	TaskList	44.4%	
	Task	100.0%	
	TaskProperties	100.0%	

6 REFERENCES

[1] http://tests.bondi.omtp.org
[2] http://coverage.omtp.org

-------- END OF DOCUMENT --------