

# BONDI

## BONDI – ARCHITECTURE & SECURITY COMMENTS TO W3C WEB APPLICATIONS WG – WIDGET DEPENDENCIES

**VERSION:** Version 1\_0  
**STATUS:** Public Working Draft  
**DATE OF LAST EDIT:** 01/08/08  
**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.

© 2008 Open Mobile Terminal Platform Ltd. All rights reserved. No part of this document may be reproduced or transmitted in any form or by any means without prior written permission from OMTP Ltd. “OMTP” and “BONDI” are registered trademarks of OMTP Ltd. Other product or company names mentioned herein may be the trademarks of their respective owners.

## **EXPRESSING WIDGET DEPENDENCY INFORMATION IN WIDGET PACKAGING**

### **INTRODUCTION**

OMTP is a forum backed by many of the largest mobile operators and has members from many hardware and software vendors across the value chain.

It was set up with the aim of gathering and driving mobile terminal requirements to ensure consistent and secure implementations, thereby reducing fragmentation and simplifying the customer experience of mobile data services. OMTP recommendations benefit carriers, content providers, middleware vendors and handset manufacturers to develop open and compatible mobile devices. OMTP have created their BONDI initiative to specifically address the problem of fragmentation in the evolution of the mobile web and to ensure the protection of the user from malicious web based services. The BONDI initiative will be delivering documentation throughout 2008 with the aim of driving activities in other appropriate standardisation and industry bodies such as W3C.

Devices supporting some of the features of BONDI will become available in 2009.

This proposal is intended as input to the W3C Web Applications work group and represents a snapshot of current discussions on including Widget Dependency Information for both heterogeneous (e.g. Mobile) and homogeneous (e.g. PC) environments in the Widget Configuration Document as defined in the W3C Widget Packaging and Configuration specification (Section 6: Configuration Document).

## TERMINOLOGY

The following terms are used in this document:

**Widget Package** – A collection of resources required by a widget (including a Widget Configuration Document) that is packaged for distributive purposes (from W3C).

**Widget Configuration Document** – an XML document that has <widget> as its root. Used to describe the configuration of a Widget Package (from W3C).

**Widget Dependency Information** – a combination of Resource Information and Device Capability Information that may be included in a Widget's Configuration Document. *Widget Dependency Information is a super-component of Resource Information and Device Capability Information.*

**Resource Information** – a collection of references and/or inferences to External Widget Resources (e.g., Widget Packages, APIs, Javascript, CSS, XSLT files and packages) that are not included in a Widget's Packaging. Resource Information can be mandatory or optional (see definitions in proposal below). *Resource Information is a sub-component of Widget Dependency Information.*

**Device Capability Information** – a collection of references and/or inferences to Device Capabilities either mandatory or optional (see definitions in proposal below) to provide the functionality of a widget. Device Capability can be hardware-based (e.g., Operating System, Provision of device features such as Camera, etc) and/or software-based (e.g., Specific Browser is available, Specific codec is available, etc). Resource Information can be included either as required (mandatory) or optional in order to provide the functionality of a widget. *Device Capability Information is a sub-component of Widget Dependency Information.*

## REQUIREMENTS

### Widget Dependency Groupings

The Widget Dependency Information required in Widget Packaging falls in to two categories:

1. **Resource Information.** External Widget Packages, Device APIs (Address Book API, Location API, etc) / Non-Device APIs (e.g. Javascript Toolkits such as Dojo, Yahoo UI, etc) and other web-based resources (e.g., CSS, XSLT, Folders) required to provide the functionality of a widget.
2. **Device Capability Information.** Minimum environment configuration required to provide the functionality of a widget (e.g., a camera, a particular sub-class of device, etc).

In addition, each component of Widget Dependency Information should either be:

1. **Mandatory** - a Resource must be present to install (if required), load and run the core features of the widget. In the case that a mandatory dependency cannot be provisioned a fatal error may occur.
2. **Optional** - a Resource may be present but could be provisioned later and/or at runtime or not at all to run peripheral / non-core functionality of the widget. In the case that an optional dependency cannot be provisioned a non-fatal error may occur.

The W3C Widget Packaging specification may wish to define how the widget handling environment will treat dependencies marked as Mandatory and Optional.

Resource Information and Device Capability Information requirements are discussed below.

## Widget Dependency Information

A widget developer should include all Widget Dependency Information in their widget configuration document.

By including this information it is foreseen that the widget developer will be improving the overall experience for their consumers by allowing the intended device to check if the required dependencies are provisioned and/or available to be provisioned prior to the provisioning of the full widget.

By specifying the Widget Dependency Information in the Configuration Document it may be possible for a consuming device to query the Configuration Document either prior to download of the entire widget package OR prior to installation/execution of a widget once it has been downloaded to a device. This may be particularly useful in environments that exhibit limited bandwidth and/or where cost is incurred in the download of data (e.g., mobile environments). It may also be used to manage user expectations that the widget will work on their intended device. A user may have the ability to check if the intended widget is likely to run on the intended device prior to incurring the cost and inconvenience of downloading, installing and/or executing the full widget package.

The Widget Dependency Information may be used by the consuming device to make decisions on which resources the widget is allowed to use. For example, access to resources declared in the Widget Dependency Information may be allowed or denied according to the security policy defined on the device. A consuming device may also allow access to resources that are requested programmatically but not declared in the Widget Dependency Information.

It is *recommended* that developers include all Widget Dependency Information in their Widget Configuration Document to aid the widget provisioning process. The security impact of omitting resources the Widget Dependency Information is currently under discussion within BONDI. For example, failure to declare a widget dependency may result in the device denying widget access to the omitted dependency.

The intention is that Widget Dependency Information in a Widget's Configuration Document shall not infer any specific means of enforcing security.

## Resource Information

It should be possible to reference multiple external Widget Resources in a Widget's Configuration Document. External Widget Resources may include Widget Packages, Javascript files, Javascript/Non-Javascript APIs, CSS files, XSLT files and Folders. The reference to any of these Widget Resources must be possible by either:

- *direct reference* to the actual location of that resource (e.g. the URL where the resource resides); OR by,
- *indirect inference* to the resource type, class or unique identifier (e.g. to allow a third-party mechanism to infer the required resource from a selection of similar resources intended for different platforms and environments).

It shall be possible to specify the version / revision number of a resource. The version or revision number can be either a single value (e.g. v1.5.3) or bounding values (e.g. v1.5.3 to v1.7) that are compatible with the intended widget.

It shall be possible to specify a friendly (display) name for each Resource Dependency required.

It shall be possible to state whether a Resource Dependency is Mandatory or Optional for installation and/or execution of the intended widget. The intention is to enable developers to programmatically provide fail-safe functionality (i.e., Exception Handling) when Optional Resource Dependencies are not present on the device. For example, if the Location API is not installed on the intended device, a Widget will still execute and is able to display maps in the widget but cannot pinpoint the user's location as part of its function.

It may be possible to indicate where a Resource Dependency should be placed in relation to the Widget Package once it has been provisioned on a device (e.g., Referred Javascript files may be required at a certain point in the relative widget directory path).

It may be possible to include a hash checksum of the external dependency for each External Resource included in the Configuration Document. This may be used by a device to check the integrity of an external dependency prior to provisioning of the external resource.

### **Device Capability Information**

It should be possible to reference multiple Device Capabilities (e.g., Device Type, Operating System, Peripheral Device Support, etc) in a Widget's Configuration Document. This may be based on the specifications provided by the W3C Ubiquitous Web Applications Working Group on Delivery Context Ontology, Interfaces and Functions or some equivalent device capability profiling initiative.

It shall be possible to specify an exact match requirement on Device Capability required by a Widget. By this we mean that it should be possible to specify whether a Device should match exactly the information provided in the Device Capability Information to ensure proper function.

In the case where an exact match is required on an *abstract* Device Capability node, a Boolean matching expression may be used to indicate if the required Device Capability is available (e.g., an intended device must have a browser = true, rather than a requirement on a specific browser being available).

It shall be possible to specify a bounding requirement on Device Capabilities required. It should be possible to specify whether the Device should match based on being greater than, less than or equal to a range of responses available (e.g., OS version is less than or equal to v8 and greater than v6).

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