mobileOK Checker 1.0a
User Manual

Version of this document: 0.6 (Draft)
Date of Release: 2007-10-16
Editors: Ignacio Marín, Abel Rionda (Fundación CTIC)
Contents

1 INTRODUCTION..........................................................................................................................4
  1.1 PURPOSE AND AUDIENCE.................................................................................................4
  1.2 COPYRIGHT ISSUES.........................................................................................................4
  1.3 STRUCTURE..........................................................................................................................5

2 DOWNLOAD AND INSTALLATION...............................................................................................6
  2.1 BINARY ARCHIVE DOWNLOAD.........................................................................................6
  2.2 SOURCE CODE DOWNLOAD (FROM CVS REPOSITORY)......................................................6

3 DESCRIPTION OF THE MOBILEOK CHECKER........................................................................6
  3.1 EXECUTION...........................................................................................................................6
    3.1.1 Prerequisites...................................................................................................................7
    3.1.2 Command line usage.....................................................................................................7
    3.1.3 Eclipse IDE usage..........................................................................................................7
  3.2 OUTPUT OF THE MOBILEOK CHECKER..............................................................................9
    3.2.1 Intermediate (mOKI) document.....................................................................................9
      3.2.1.1 primaryDoc element...............................................................................................9
      3.2.1.2 stylesheets element...............................................................................................12
      3.2.1.3 images element.......................................................................................................12
      3.2.1.4 links element..........................................................................................................13
    3.2.2 Results document..........................................................................................................14

4 EXAMPLES........................................................................................................................................16
  4.1 EXAMPLE 1...........................................................................................................................16
  4.2 EXAMPLE 2...........................................................................................................................18
  4.3 EXAMPLE 3...........................................................................................................................18
  4.4 EXECUTING JUNIT TESTS....................................................................................................18

5 KNOWN ISSUES..........................................................................................................................19
  5.1 ABOUT W3C MOBILEOK CHECKER 1.0A..............................................................................19
  5.2 ABOUT THIS USER MANUAL..............................................................................................19

6 REFERENCES..................................................................................................................................20

7 ACKNOWLEDGEMENTS...............................................................................................................20
Illustration Index

Illustration 1: Setting a configuration to execute the Checker in Eclipse IDE.....................................................8
Illustration 2: Passing parameters to the Checker in Eclipse IDE...........................................................................8
Illustration 3: Main elements in a mOKI document.............................................................................................9
Illustration 4: Children elements of primaryDoc...............................................................................................9
Illustration 5: Information in an HTTPRequest element....................................................................................10
Illustration 6: Example of representation of elements in an HTTP Request header............................................10
Illustration 7: Example of HTTPResponse element..........................................................................................11
Illustration 8: Example of XHTMLDocInfo element..........................................................................................12
Illustration 9: Example of information contained in images element.................................................................13
Illustration 10: Example of links element...........................................................................................................14
Illustration 11: results.xml example................................................................................................................15
Illustration 12: Details for CACHING test which PASSes with WARNings.........................................................15
Illustration 13: Naming of subtests in mobileOK Basic Tests 1.0 document.........................................................16
Illustration 14: Additional information for FAILing results..................................................................................16
Illustration 15: Gathering essential information from results.xml for example 1..............................................17
Illustration 16: Finding causes for CACHING and CONTENT_FORMAT_SUPPORT warns in moki.xml for example 1.................................................................17
Illustration 17: Execution of a JUnit test.............................................................................................................18
Illustration 18: Arguments for the JUnit test.......................................................................................................19
1 Introduction

1.1 Purpose and Audience

This document is meant to serve as a User Manual for the W3C mobileOK Checker. Current version of this User Manual (0.5) is a draft, so the editors intend to write new versions correcting typos and errors and inserting new and more complete and accurate information.

The version of mobileOK Checker subject to comments in this document is 1.0a, which is an Alpha release of the checker. A Beta version of this software and a new version of this user manual is expected by next W3C Technical Plenary week in 2007.

The audience of this document is the general public being aware of the need to check conformance of web resources (and also of the way in which they are delivered) to W3C mobileOK Tests 1.0. W3C mobileOK Checker is a software tool, created by means of collaboration among different organizations (see Acknowledgements), with the intention to create a Reference Implementation so the W3C mobileOK Tests 1.0 document can become a W3C Recommendation. W3C mobileOK Checker 1.0a allows users to pass mobileOK Tests on a given web resource.

W3C mobileOK Basic tests implement the way in which some of the Best Practices in the W3C Mobile Web Best Practices (Basic Guidelines) 1.0 document can be checked against web resources. The term “some” refers to all those Best Practices that can be verified by a machine (i.e., which can be described as an algorithm). Those Best Practices which can not be described as an algorithm (needing the assessment of a human expert) are covered by the future W3C Mobile OK Pro Tests document, still in development.

The next version of this document will be accompanied by a W3C mobileOK Checker 1.0a Developer Manual. While this User Manual tries to explain the W3C mobileOK Checker 1.0a software tool as a black box and to cover all the subjects of interest to anyone trying to use the tool as-is, the Developer Manual describes the internal design and implementation details of the tool as it would be interesting for developers willing to modify, extend or enhance the functionality of the tool or even willing to integrate the tool with their own software or services.

The W3C mobileOK Checker 1.0a, its User Manual and Developer Manual, the W3C mobileOK Tests 1.0 and the W3C Mobile Web Best Practices (Basic Guidelines) 1.0 are efforts taking place under the framework of the W3C Mobile Web Best Practices working group (part of the W3C Mobile Web Initiative).

1.2 Copyright issues

W3C mobileOK Checker 1.0a software tool and related documents (User Manual and Developer Manual) are released under the W3C License, in its current version at the very moment of the release of this document.

Several external Java libraries and software tools are used by the W3C mobileOK Checker. Their use is subject to different licenses and copyright clauses. The binary distribution for the W3C mobileOK Checker and its source code in W3C’s CVS repository contains the licenses and copyright claims to which each of them are subjected.

Eclipse, the IDE used as example in this documentation, is released under the Eclipse Public License v1.0 in the version used for the screenshots taken.
1.3 Structure

This document is divided into several sections, which are now briefly described:

- **Introduction**: Current section, where the general purpose of this document and of the target software tool (W3C mobileOK Checker 1.0a) is commented, including related W3C activities and documents.

- **Download and installation**: This section describes how to download the Java executable binary file (in JAR format) for the W3C mobileOK Checker, how to optionally download the source files for the software and the library dependencies for the Checker.

- **Description of the mobileOK Checker**: This section explains the prerequisites needed for the software to work, how to execute the W3C mobileOK Checker JAR file from command line or from Eclipse IDE, how to run some specific tests included in the distribution and also the format of the output documents that the Checker generates.

- **Examples**: This section includes several examples meant to illustrate how to execute the Checker and how to interpret the documents produced.

- **Known issues**: Issues regarding known bugs and future improvements of the Checker, and missing information (that will be included in future versions) in this User Manual.

- **References**: Documents interesting for readers, which might help them to understand this document, the W3C mobileOK Checker 1.0a software tool and all related work of the W3C Mobile Web Best Practices working group.

- **Acknowledgements**: Credits to the people who helped developing this User Manual and the W3C mobileOK Checker 1.0a software tool.
2 Download and installation

2.1 Binary archive download.

The Java binary executable JAR file is downloadable at http://dev.w3.org/cvsweb/~checkout~/2007/mobileok-ref/mobileOK-Basic-RI-1.0-deploy.jar?rev=1.1&only_with_tag=HEAD.

2.2 Source code download (from CVS repository)

The source files corresponding to the JAR file mentioned in 2.1 are available in W3C's CVS repository at :pserver:anonymous@dev.w3.org/public/sources/2007/mobileok-ref (password for anonymous user is anonymous). If previous URL does not work, the specific data for CVS access are commented below:

- Host: dev.w3.org
- Repository path: /sources/public
- User: anonymous
- Password: anonymous
- Connection type: pserver
- Port: default pserver port (2401)
- Specified module name: 2007/mobileok-ref.
- Available branches: Only HEAD. No other branches expected in the future.

Please take into account that, in order to be able to deal with the source code as a developer, you will need to read the W3C mobileOK Checker Developer Manual document (still in development).

3 Description of the mobileOK Checker

This section is the core of this document and tries to help users to be able to execute the Checker and perform a mobileOK Basic test (or a set of tests, or even all mobileOK tests) on a given web resource (specified by a URI).

For a more in-depth analysis of the source code, the W3C mobileOK Checker 1.0a Developer Manual will be available when next version of this document is published.

3.1 Execution

In addition to command-line execution, an example using Eclipse IDE is showed. Before using any IDE, it is important to read subsection 2.3 of this document explaining command-line usage of the W3C mobileOK Checker.
3.1.1 Prerequisites

This library was made using Java 1.5. In order to avoid compatibility problems, Java 1.5 or a higher version should be installed in your system and made available to the Checker JAR file.

3.1.2 Command line usage

In order to execute the library from command line, the format to launch the Checker is as shown below:

```java
java -jar mobileOK-Basic-RI-1.0-deploy.jar URL [test] [options]
```

where:

- **URL**: The web resource to check against mobile OK Basic Tests. For example `http://www.example.org`.
- **[test]**: An optional argument to check the web resource against a specific mobileOK Basic Test. By default all mobileOK Basic Tests are checked. The format name of the tests is the same of mobile OK Basic Tests document (based on Mobile Web Best Practices 1.0).

  For example, this would be the way to check the example site against MEASURES test:
  ```
  java -jar mobileOK-Basic-RI-1.0-deploy.jar http://www.example.org MEASURES
  ```

- **[options]**: Some options related to the output can be set. By default the common output stream is used.
  - `-r filename`: Results are saved in the specified file.
  - `-m filename`: The (mOKI, mobileOK Intermediate) document is saved in the specified file.

3.1.3 Eclipse IDE usage

After source code is downloaded from W3C's CVS repository and a new Eclipse Java project is created (see 2.3), it is possible to define a run configuration for the library.

- Launch a new run configuration (Run option in Run menu).
- Introduce as main class `org.mwi.mobileok.basicTester`
Introduce, as program arguments, the URI of the web resource to be checked and the rest of the options (as shown in 3.1.2). For example in the following screenshot, MEASURES test is tested against http://www.mysite.com. Furthermore, the results are save in a file called results.xml.

Illustration 1: Setting a configuration to execute the Checker in Eclipse IDE

Illustration 2: Passing parameters to the Checker in Eclipse IDE

Finally, the Run button must be activated to launch the new configuration.
3.2 Output of the mobileOK Checker

When tests are performed by the W3C mobileOK Checker on a web resource, two documents are generated. After retrieving the web resource to be examined, a preprocessing stage is performed by the Checker and an intermediate document is generated. This document is then processed in order to generate the final results document, where the output of the tests is stated. The internal architecture for the W3C mobileOK Checker 1.0a will be detailed in the Developer Manual.

3.2.1 Intermediate (mOKI) document

The format for the intermediate document generated by the Checker (named as mOKI format by the authors of the library, which stands for mobileOK Intermediate format) is explained in this subsection. This document tries to describe the context in which the web resource to be checked has been retrieved. This context is basically the HTTP dialog taking place between the Checker and the server in order to retrieve the web resource, including HTTP requests, HTTP responses, the content of the web resource as delivered by the server in one of the responses and, of course, all the HTTP headers included in requests/responses -which are really important for some tests like CACHING-.

After, the XML declaration, a moki element is the root element of this XML document dialect. Inside the root element, a sequence of four elements appears in a mOKI document: primaryDoc, stylesheets, images and links. Note that no XML Schema or DTD is included. By now, none of them has been created for mOKI format but it is envisaged the creation of a DTD in the near future so mOKI documents can be validated against the mOKI DTD.

Illustration 3: Main elements in a mOKI document

3.2.1.1 primaryDoc element

This element intends to represent the primary document that the Checker needs to retrieve in order to perform the test. This representation includes retrieval information (element retrieval, child of primaryDoc) and information related to the XHTML document retrieved -when the URI on which to pass the test references an XHTML document- (XHTMLDocInfo child element).

Illustration 4: Children elements of primaryDoc

3.2.1.1.1 retrieval element

This element includes the URI of the retrieved web resource (child element retrievedURI) and a sequence of pairs of HTTPRequest and HTTPResponse elements, which represents the HTTP dialog that took place in order to obtain the web resource being target of the validation against mobileOK Basic Tests 1.0.
3.2.1.1.1 HTTPRequest element

An HTTPRequest element holds the raw headers of a request, the HTTP method used for the request (GET, POST, HEAD), the URI requested, the HTTP protocol version, and a list of header elements representing the HTTP headers included in the request. Instead of describing each child element of HTTPRequest, a visual example is shown below in order to clarify the contents of an HTTPRequest element.

```xml
<HTTPRequest>
  <method>GET</method>
  <uri>http://www.example.com/xhtml</uri>
  <protocol.HTTP/1.1/>
  <header name="User-Agent" value="W3C-mobileOK/DDC-1.0 (see http://www.w3.org/2006/07/mobileok-ddc)"/>
  <header name="Host" value="www.example.com"/>
  <header name="Accept-Charset"/>
</HTTPRequest>
```

Illustration 5: Information in an HTTPRequest element

Note that, although a header element keeps the information of an HTTP Request header by means of a name attribute and a value attribute in Illustration 5, this is not true in all cases, as shown in Illustration 6.

```xml
<header name="accept">  
  <element name="application/xhtml+xml"/>
  <element name="text/html">  
    <parameter name="q" value="0.1"/>
  </element>
  <element name="application/vnd.wap.xhtml+xml">  
    <parameter name="q" value="0.1"/>
  </element>
  <element name="text/css"/>
  <element name="image/jpeg"/>
  <element name="image/gif"/>
</header>
```

Illustration 6: Example of representation of elements in an HTTP Request header

Illustration 6 shows how the use of name/value is not sufficient to express the information contained in HTTP Request headers as the Accept header, so a different model with element children and parameter children to element is used. This way, headers including a set of values and a qualifier for each value can be easily expressed.

3.2.1.1.2 HTTPResponse element
This element is analogous to the `HTTPRequest` element in mOKI. It includes the raw version of the information contained in the HTTP Response, the `protocol` version in use, the `status` code for the response (and its corresponding reason string), a sequence of `header` elements (analogous in purpose and format to the element of the same name which is children of `HTTPRequest`) and an `entity` element including the body of the HTTP Response which replaces different characters (’<’, ’>’,....) by their corresponding XML entities.

```
<HTTPRequest>
  <crlfHeaders>Set-Cookie: PREF=ID=1fc84a343529b7f:7f:TM-1184686864:cLM-1184686864:S=74E46aD52D0K39e1;
  expires=Sun, 17-Jan-2030 19:14:07 GMT; path=/; domain=.example.com;82;

  <protocol HTTP/1.1
  <status code="200" reason="OK"/>
  <header name="set-cookie" value="HPRF-MisIAAAALAAAAAAAANAAAAAAAAL;
  expires=Sun, 17-Jan-2030 19:14:07 GMT; path=/;
  domain=.example.com"/>

  <header name="date" value="Tue, 17 Jul 2007 13:41:04 GMT"/>
  <header name="pragma" value="no-cache"/>
  <header name="content-type">
  <header name="transfer-encoding" value="chunked"/>
  <header name="cache-control">

  <entity size="1106">&lt;?xml version="1.0" encoding="UTF-8"?&gt;
</entity>  
</HTTPResponse>
```

Illustration 7: Example of HTTPResponse element

3.2.1.1.2 XHTMLDocInfo element

This element includes information about the XHTML document targeted by the Checker for its validation against W3C mobileOK Tests 1.0. As shown in Illustration 8, it contains information about the validation of the document:

- **UTF-8 validity**: true, if all characters in the document are expressed in UTF-8, or false (if not).
- **Markup validity**: true, if the document validates against its DTD; false, if not.
- **Mobile validity**: true, if the document validates against XHTML Basic or XHTML Mobile Profile DTDs; false, if not.

It also provides the number of extraneous characters (extra spacing characters) and the total number of characters, information about the XML declaration and the DOCTYPE declaration and, finally, a `docContent` element containing the XHTML document itself.
The example above also shows how the mOKI format expresses errors when validity against given criteria are not met. Errors are expressed as a sequence of error elements (children of the UTF-8Validity, MarkUpValidity or MobileValidity elements) and include the reason of the error (info element) and its position in the document.

### 3.2.1.2 stylesheets element

This element includes CSS information relative to the XHTML document target of the mobileOK Basic 1.0 tests. It consists of a sequence of stylesheet elements, each of them containing a URI element (representing the URI for a external stylesheet, or the URI of the document itself if the CSS is inline or embedded), a CSSValidity element indicating if the style information is CSS-valid and a styles element.

The styles element is a sequence of style elements: each style element represents a CSS style sentence in XML format.

If CSSValidity element indicates that the stylesheet is not valid, the element will include a sequence of error elements, analogous to the error elements seen in XHTMLDocInfo element.

### 3.2.1.3 images element

This element intends to express information related to the images referenced by the XHTML document. It consists of a sequence of image elements, each of them including the URI of the image, information about the process of retrieval of the image (retrievedURI, HTTPRequest and...
MobileOK Checker 1.0a

HTTPResponse mOKI elements already explained in previous sections), and information of the image of special interest to mobile OK Basic Tests 1.0 (imageInfo element).

3.2.1.4 links element

This element includes information relative to other web resources (others than external stylesheets and images) referenced in the XHTML document. This element has as many link children elements as external web resources are referenced in the document. Each link element contains the URI of the link and retrieval information (HTTP dialog to retrieve the web resource corresponding to the URI).
Results document

`results.xml` is the final document generated by W3C mobileOK Checker 1.0a after processing the previously generated mOKI document. It consists of an XML root element named `tests` which contains a sequence of `test` elements. The `test` element will be qualified by an `outcome` attribute. The value of `outcome` can be “FAIL” (if any of the `test` children elements have an `outcome` attribute with a value of “FAIL”) or “PASS” (if all the `test` children elements have an `outcome` attribute with a value of “PASS”). For an explanation of the formal meaning given to “FAIL” and “PASS” (and “WARN”), and for the definition and naming of each test, it is recommended to visit the W3C mobileOK Tests 1.0 document. Thus, `results.xml` will indicate in `tests` element that all the tests passed (or that there is a warning for the user to take into account, at most) or that any of the tests failed. Remember that the tests performed by the Checker should have been given to the JAR binary file as an optional second argument, just after the URI to be checked (see subsection Execution of this document). If no test is passed as argument to the Checker, all the tests will be performed against the URI.
The example in Illustration 11 shows test results for the mOKI document used in the previous subsection of this document: Intermediate (mOKI) document. Note how the name given to each test is the same name given to tests in the W3C mobileOK Tests 1.0 document.

Each test element will include a sequence of result children elements if its outcome is “FAIL” or if it is “PASS” but the checker needs to warn the user about any interesting issue related to the test.

Illustration 12 shows how a test that passed successfully includes extra results information to want the user about certain interesting issues. It is important to note that the name attribute for each result element with a test element references each subtest being part of a test, as mentioned in W3C mobileOK Tests 1.0 document (see Illustration 13).
3.2 CACHING

The purpose of the test is to alert providers to the fact that their content may not be cached, if it would be beneficial to do so.

Note:
Where both a meta element with http-equiv attribute and the corresponding HTTP header are found, the value of the HTTP header must be used - see also note under 2.4.4 Meta http-equiv Elements

![Illustration 14: Additional information for FAILing results](image)

Results information related to failing tests or tests with warnings include general information (info element containing the text of the subtest in the W3C mobileOK Tests 1.0 document), the origin of the error (code element, which generally reports the markup causing problems and/or the output error of an underlying library used by the checker), and the position of the error (as seen in Illustration 14).

4 Examples

This section tries to illustrate the use of the W3C mobileOK Checker 1.0a with real mobile sites (although real URIs have been changed following the example.domain model). The only intention of this section is to help readers to execute the Checker and interpret its results.xml output document.

4.1 Example 1

First example is a test that fails on CONTENT_FORMAT_SUPPORT and VALID_MARKUP, while it warns about some issues regarding CACHING, DEFAULT_INPUT_MODE and STYLE_SHEETS_USE. At first place, it would be necessary to examine the results.xml document (shown in Illustration 16) to gather all the information needed to infer the previous conclusions.
Now, moki.xml would allow the user to confirm what results.xml indicates:

Illustration 15: Gathering essential information from results.xml for example 1

Illustration 16: Finding causes for CACHING and CONTENT_FORMAT_SUPPORT warns in moki.xml for example 1
4.2 Example 2

4.3 Example 3

4.4 Executing JUnit Tests

The W3C mobileOK Checker has been developed using JUnit Test Framework. This section explains how to launch the JUnit tests created in the project, also included in the JAR binary release and in the source code in the CVS. It is important that readers are aware of the fact that creation and execution of unitary tests will be covered in the W3C mobileOK Checker 1.0a Developer Manual (still in development). All the steps followed in this example have been done by using the Eclipse IDE.

- Launch a new run configuration (Run option in Run menu).
- Introduce as main class org.w3c.mwi.mobileok.basic.OneTestTest

Illustration 17: Execution of a JUnit test

- Enter, as program arguments, one of the mobileOK tests in the format specified in 3.1.2. The JUnit test for the specified mobileOK test will be launched. If no mobileOK test is specified, all tests will be launched.
5 Known issues

5.1 About W3C mobileOK Checker 1.0a

The development of the tool is alive. Please do not hesitate to send your comments to the mobileOK Checker Task Force mailing list (public-mobileok-checker@w3.org). Your feedback is very important for the authors of W3C mobileOK Checker 1.0a so this software can be improved. Optionally, you can access the Checker's bug tracker at http://www.w3.org/Bugs/Public/ and help W3C to develop this tool.

5.2 About this User Manual

- **General issues**: Review and improve wording all across the document. Develop the glossary.
- **Subsection 2.3**: Dependencies: Instead of commenting dependencies of the Checker as a list of JAR files, future versions of this User Manual will categorize dependencies by libraries or tools, grouping them so the subsection is more clear and brief.
- **Subsection 3.2**: More detailed description of mOKI and result formats.
- **Section 4 (Examples)**: Detail Example 1 in more depth. Add Examples 2 and 3.
6 References

NOTE: When referring a W3C document, the version of the document that has been considered at this version of the W3C mobileOK Checker User Manual appers in this section. Additionally, the URI to the latest version of such document is offered between square brackets.

1) W3C mobileOK Tests 1.0, [http://www.w3.org/TR/2007/WD-mobileOK-basic10-tests-20070928/]. Editors: Sean Owen (Google) and Jo Rabin (mTLD) [latest version: http://www.w3.org/TR/mobileOK-basic10-tests/].

2) W3C Mobile Web Best Practices (Basic Guidelines) 1.0, [http://www.w3.org/TR/2006/PR-mobile-bp-20061102/]. Editors: Jo Rabin (mTLD) and Charles McCathieNeville -early drafts- (Opera Software) [latest version: http://www.w3.org/TR/mobile-bp/].

3) W3C License, [http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231].

4) W3C Mobile Web Best Practices working group, [http://www.w3.org/2005/MWI/BPWG/].

5) W3C Mobile Web Initiative, [http://www.w3.org/Mobile/].

7 Acknowledgements

W3C mobileOK Checker 1.0a is the result of the effort of several organizations. In alphabetical order, the members of the W3C mobileOK Checker Task Force (taking place within the W3C Mobile Web Best Practices working group) who contributed to the creation of this software tool and its documentation are:

- Roland Gülle (Sevenval)
- Miguel García (Fundación CTIC)
- Dominique Hazael-Massieux (W3C)
- Laura Holmes (Google)
- Ignacio Marín (Fundación CTIC)
- Ruadhan O'Donoghue (mTLD)
- Sean Owen (Google, chair of the Task Force)
- Jo Rabin (mTLD)
- Abel Rionda (Fundación CTIC)