WML Tag Library Reference

Openwave™ Usability Interface, Java Edition 1.0 Beta

Openwave Systems Inc.
1400 Seaport Boulevard
Redwood City, CA 94063
http://www.openwave.com

Part Number OUWT-10-004
October 2001

LEGAL NOTICE

Copyright © 1994–2001, Openwave Systems Inc.
All rights reserved.

Openwave, the Openwave logo, Phone.com, the Phone.com logo, and the family of terms carrying the “UP.” prefix are trademarks of Openwave Systems Inc.

All other company, brand, and product names are referenced for identification purposes only and may be trademarks that are the sole property of their respective owners.

Contents

About This Book xi
About OUI xi
Related Documentation xii
Technical Support xii
Other Resources xiii
WAP xiii
Java xiii

1 WML Tag Library Reference 1
Common Attributes 1
<a> 1
Description 1
Attributes 2
Syntax 2
Example 2
<access> 3
Description 3
Attributes 3
Syntax 3
<additional_format> 4
Description 4
Attributes 4
Syntax 4
Example 5
<additional_src> 6
Description 6
Attributes 6
Syntax 6
Example 7
<anchor> 8
Description 8
Attributes 8
Syntax 8
Example 9

<appendix> 10
Description 10
Attributes 10
Syntax 10
Example 10
<b> 11
Description 11
Attributes 11
Syntax 11
<big> 11
Description 11
Attributes 11
Syntax 11
<body_pager> 12
Description 12
Attributes 12
Syntax 13
Example 14
<bp_chunk> 15
Description 15
Attributes 15
Syntax 15
Example 16
<bp_footer> 17
Description 17
Attributes 17
Example 17
<bp_header> 18
Description 18
Attributes 18
Example 18
<bp_percentage> 19
Description 19
Attributes 19
Syntax 19
Example 19

 20
Description 20
Attributes 20
Syntax 20
<button> 21
Description 21
Attributes 21
Syntax 21
Example 22

<caller> 23
Description 23
Attributes 23
Syntax 23
Example 24
<card> 25
Description 25
Attributes 26
Syntax 27
Examples 29
<check> 33
Description 33
Attributes 33
Syntax 34
Example 34
<combo_menu> 35
Description 35
Attributes 35
Syntax 35
Example 36
<condition> 37
Description 37
Attributes 37
Syntax 37
Example 38
<conditionlist> 39
Description 39
Attributes 39
Syntax 39
Example 40
<do> 41
Description 41
Attributes 41
Syntax 43
<else> 44
Description 44
Syntax 44
Example 44
 45
Description 45
Attributes 45
Syntax 45
<fieldset> 46
Description 46
Attributes 46
Syntax 46

<form> 47
Description 47
Attributes 47
Syntax 48
Example 49
<go> 50
Description 50
Attributes 50
Syntax 51
<head> 52
Description 52
Syntax 52
Example 52
<hr> 53
Description 53
Attributes 53
Syntax 53
Example 54
<if> 55
Description 55
Attributes 55
Syntax 56
Example 57
<img> 58
Description 58
Attributes 58
Syntax 59
Example 61
<input> 62
Description 62
Attributes 62
Syntax 64
Example 66
<menu> 67
Description 67
Attributes 67
Syntax 67
Example 67
<menu_item> 68
Description 68
Attributes 68
Syntax 69
Example 70

<meta> 71
Description 71
Attributes 71
Syntax 71
Example 72
<method> 73
Description 73
Attributes 73
Example 73
<noop> 73
Description 73
Syntax 73
<onevent> 74
Description 74
Attributes 74
Syntax 74
<optgroup> 76
Description 76
Attributes 76
Syntax 76
<option> 77
Description 77
Attributes 77
Syntax 77
Example 77
<p> 78
Description 78
Attributes 78
Syntax 78
<param> 79
Description 79
Attributes 79
Syntax 79
Example 79
<pre> 80
Description 80
Attributes 80
<picker> 81
Description 81
Attributes 81
Syntax 82
Example 83
<picker_card> 84
Description 84
Attributes 84
Syntax 85
Example 86

<popup> 87
Description 87
Attributes 87
Syntax 88
Example 88
<postfield> 89
Description 89
Attributes 89
Syntax 89
<prefetch> 90
Description 90
Attributes 90
Syntax 90
Example 91
<prev> 91
Description 91
Attributes 91
Syntax 91
<primary_path> 92
Description 92
Attributes 92
Syntax 92
Example 93
<radio> 94
Description 94
Attributes 94
Syntax 95
Example 95
<raw_mode> 96
Description 96
Attributes 96
Syntax 97
Example 98
<refresh> 99
Description 99
Attributes 99
Syntax 99
<rendering_directive> 100
Description 100
Attributes 100
Syntax 100
Example 102
<secondary_path> 104
Description 104
Attributes 104
Syntax 104
Example 105

<setvar> 106
Description 106
Attributes 106
Syntax 106
<side_path> 107
Description 107
Attributes 107
Syntax 107
Example 108
<small> 108
Description 108
Attributes 108
Syntax 108
 109
Description 109
Attributes 109
Syntax 109
<table> 110
Description 110
Attributes 110
Syntax 111
Example 111
<task_menu> 114
Description 114
Attributes 114
Syntax 114
Example 115
<tc> 116
Description 116
Attributes 116
Syntax 116
Example 116
<td> 117
Description 117
Attributes 117
Syntax 117
Example 117
<template> 118
Description 118
Attributes 118
Syntax 118
<th> 119
Description 119
Attributes 119
Syntax 119
Example 119

<then> 120
Description 120
Syntax 120
Example 120
<timer> 121
Description 121
Attributes 121
Syntax 121
<tr> 122
Description 122
Attributes 122
Syntax 122
Example 122
<u> 122
Description 122
Attributes 122
Syntax 122
<unset> 123
Description 123
Attributes 123
Syntax 123
Example 123
<wml> 126
Description 126
Attributes 126
Syntax 126
Example 127

About This Book

This book contains reference information for the Openwave Usability Interface (OUI), Java Edition 1.0 Beta WML Tag Library. The OUI WML Tag Library provides access to OUI features through a familiar WML-based tag interface.

About OUI

OUI was developed out of the need for developers to build a single application that runs properly across multiple browsers on multiple mobile devices. Openwave is committed to providing the best WAP development tools and created OUI to help developers build best-of-breed applications with the least amount of work.

OUI is a software component that you can download without charge from Openwave and use with your current web development platform to build usable WAP applications.

Currently OUI is designed for the Java platform, for use with Java Servlets, JSP, and other Java-based environments

Refer to the OUI *Getting Started* book for more information about OUI.

Related Documentation

OUI comes with the following documentation:

- *Installation and Integration* describes how to install OUI along with Java server requirements. It also includes a basic OUI example.
- *Getting Started* contains an overview of OUI and how to get started writing OUI applications.
- The *Object Model Reference* describes the details of Java implementation for each OUI object.
- The *WML Tag Library Reference* (this book) describes the OUI WML tag library. The tags and their attributes are described in a reference format.
- The *XHTML Tag Library Reference* describes the XHTML tag library. The tags and their attributes are described in a reference format.

The Openwave SDK comes with documentation for XHTML-MP, WML and WML script, and related topics. For a complete list of documentation, see:

`http://developer.openwave.com`

Technical Support

The best resource for up-to-date information on using OUI is the Openwave Developer site:

`http://developer.openwave.com`

In addition to the downloadable OUI, this site contains a variety of useful resources, including Frequently Asked Questions, bug reporting, technical support, and an interactive developer forum.

Other Resources

WAP

- WAP FAQs: http://www.allnetdevices.com/faq/
- WAP Forum: http://www.wapforum.com
- WirelessDeveloper.com: http://www.wirelessdeveloper.com
- WirelessDevNet.com: http://www.wirelessdevnet.com
- WML Forum: http://groups.yahoo.com/group/wmlprogramming
- WML Pulse Europe: http://wmlpulse-europe.openwave.com
- WMLScript.com: http://www.wmlscript.com

Java

- Allaire JRun: http://www.jrun.com/
- Jakarta Tomcat: http://jakarta.apache.org/
- Resin-CMP: http://www.caucho.com
- SUN JSP: http://java.sun.com/products/jsp
- SUN Servlets: http://java.sun.com/products/servlet

1 WML Tag Library Reference

This chapter is a reference to the OUI WML tag library. It contains detailed descriptions of each tag and its attributes, listed in alphabetical order.

Common Attributes

Unless otherwise specified, the following attributes are common to all tags.

Table 1-1. Common attributes

Name	Description
`xml:lang`	Specifies the natural language for the element and its contents.
`id`	Specifies the name of the element.
`class`	Specifies the class name for the element.

<a>

Description

The OUI <a> tag encapsulates the WML <a> element. Like the WML <a> element, it specifies a link to another card, deck, or other resource. The main purpose of encapsulating this element is to render the `accesskey` attribute only for browsers that support it. The `accesskey` attribute is a WML extension that you can use to associate a number between 0 and 9 with the link. This is a phone key pad accelerator so that users can trigger the link simply by pressing the corresponding number.

This tag is the short syntax form for anchors, but it can only be used to define implied <go> tasks that require a URL specification.

In general, you should use the <a> tag rather than the combination of the <anchor> and <go> tags because the <a> tag is shorter, easier to optimize, and easier to write.

Attributes

Table 1-2. <a> tag attributes

Name	Required	rtexprvalue	Description
`title`	No	true	A label that identifies the link. If you do not specify the `title` attribute, the device uses the word “Link” as the default label.
`accesskey`	No	true	A number (0-9) that appears on the left side of the screen next to the link. When the user presses the corresponding key on the phone keypad, the phone executes the task defined by the link.
`href`	Yes	true	Specifies the URL to open when the link is activated.

Syntax

```
<oui:a href="url" title="label" accesskey="number" >
    ...any valid combination of text, <oui:img>, and <oui:br/>
        elements
</oui:a>
```

Example

The following example shows how to use the <a> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="anchorsample" title="Access Keys">
        Sample links
        <oui:p>
            <oui:a href="#tbox1" title="tbox1"
                accesskey="1">Timebox1</oui:a>
            <oui:a href="#tbox2" title="tbox2"
                accesskey="2">Timebox2</oui:a>
            <oui:a href="#tbox3" title="tbox3"
                accesskey="3">Timebox3</oui:a>
        </oui:p>
    </oui:card>
</oui:wml>
```

<access>

Description

The <access> tag encapsulates the WML <access> element. You can use it to limit access to decks originating from the specified domain name and path.

Attributes

Table 1-3. <access> **tag attributes**

Name	Required	rtexprvalue	Description
domain	No	true	The URL domain of other decks that can access cards in the deck. The default value is the domain of the current deck.
path	No	true	The root URL of other decks that can access cards in the current deck. The default value is "/" (the root path of the current deck), which allows any deck within the specified domain to access this deck.

Syntax

```
<oui:head>
    <oui:access domain="domain" path="path"/>
    ...
</oui:head>
```

<additional_format>

Description

You can use the <additional_format> tag to specify a different input format for a particular browser, family of browsers, or gateway. This is useful for devices that do not provide adequate feedback to the user about the format for an input field.

Attributes

Table 1-4. <additional_format> tag attributes

Name	Required	rtexprvalue	Description
useragent	No	true	A string that identifies the browser. OUI performs a partial match of this string to the contents of one of the HTTP_USER_AGENT headers from an HTTP request. The user agent header includes the string "UP.Browser" for Openwave browsers using Openwave gateways, and it includes the string "Nokia" for Nokia browsers. In most cases, Openwave browsers using non-Openwave gateways include the string "UP/".
agentfamily	No	true	UPText | UPGUI | Nokia | MSIE5 | Generic Identifies the family of the browser being used.
gateway_vendor	No	true	Openwave | Nokia | CMG Identifies the vendor of the gateway.
version	No	true	Identifies the browser version.
format	Yes	true	Specifies a data format that the user entry must match.

Syntax

```
<oui:additional_format format="format or mask"
                       useragent="useragent or browser"
                       agentfamily="family of browsers"
                       gateway_vendor="vendor"
                       version="browser version" />
```

Example

The following example shows how to use the <additional_format> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="formatsample" title="Addtnl Format">
        <oui:p>
        Phone no:
            <oui:input type="text" name="phoneno" value=""
                format="AAAAAAA">
                <oui:additional_format agentfamily="UPText"
                    format="\\2\\0\\2\\-NNNNNNN" />
                <oui:additional_format agentfamily="UPGUI"
                    format="\\(\\2\\0\\2\\)NNN\\-NNNN" />
                <oui:additional_format agentfamily="Nokia"
                    format="NNNNNNN" />
                <oui:additional_format agentfamily="MSIE"
                    format="NNN\\XNNNNNNN" />
            </oui:input>
        </oui:p>
    </oui:card>
</oui:wml>
```

NOTE To include a backslash as part of a string literal, you must precede it with another backslash. This is commonly referred to as an *escape sequence*.

<additional_src>

Description

You can use the <additional_src> tag to specify a different image for a particular browser, family of browsers, or gateway. This enhances the usability of your application by using different image formats (such as *.wbmp, *.png, and so on) depending on the formats supported by the device.

Attributes

Table 1-5. <additional_src> **tag attributes**

Name	Required	rtexprvalue	Description
useragent	No	true	A string that identifies the browser being used. This string is compared to the contents of the HTTP_USER_AGENT header from an HTTP request. The user agent header includes the string "UP.Browser" for Openwave browsers using Openwave gateways, and it includes the string "Nokia" for Nokia browsers. In most cases, Openwave browsers using non-Openwave gateways include the string "UP/".
agentfamily	No	true	UPText | UPGUI | Nokia | MSIE5 | Generic Identifies the family of the browser being used.
gateway_vendor	No	true	Openwave | Nokia | CMG Identifies the vendor of the gateway.
version	No	true	Identifies the browser version.
src	Yes	true	Specifies the image to be used if the conditions are satisfied

Syntax

```
<oui:additional_src src="URL of the image"
                    useragent="useragent or browser"
                    agentfamily="browser family"
                    gateway_vendor="gateway vendor"
                    version="browser version" />
```

Example

The following example shows how to use the <additional_src> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="card1" title="IMG Test" >
        <oui:p align="center" mode="wrap">
            This tests the additional_src tag.
            Here is the image:
            <oui:img alt="UK Compass"
                src="http://www.openwave.com/images/
                default_logo.wbmp">
                <oui:additional_src agentfamily="UPGUI"
                    src="http://www.openwave.com/images/
                    opwv.png" />
                <oui:additional_src agentfamily="UPText"
                    src="http://www.openwave.com/images/
                    opwv.wbmp"/>
            </oui:img>
        </oui:p>
    </oui:card>
</oui:wml>
```

<anchor>

Description

The <anchor> tag encapsulates the WML <anchor> element. Like the WML <anchor> element, it specifies a link to another card, deck, or other resource. The main purpose of encapsulating this element is to render the accesskey attribute only for browsers that support it. The accesskey attribute is a WML extension that you can use to associate a number between 0 and 9 with the link. This is a phone key pad accelerator so that users can trigger the link simply by pressing the corresponding number.

Attributes

Table 1-6. <anchor> tag attributes

Name	Required	rtexprvalue	Description
title	No	true	A label that identifies the link. If you do not specify the title attribute, the device uses the word “Link” as the default label.
accesskey	No	true	A number (0-9) that appears on the left side of the screen next to the link. When the user presses the corresponding key on the phone keypad, the phone executes the task defined by the link.

Syntax

```
<oui:anchor title="label" accesskey="number">
    task text
</oui:anchor>
```

In the syntax for this tag, `task` represents the action to perform when the user activates the link, and `text` is the text the device displays to represent the link.

Table 1-7. <anchor> tag syntax elements

`task`	You must anchor one of the following task elements to a link: • `<oui:go>` • `<oui:prev>` • `<oui:refresh>` • `<oui:caller>` You can also anchor the Openwave extensions (`<spawn>`, `<exit>`, and `<throw>` tasks) using the `<raw_mode>` tag. These tasks are not directly supported because they cannot be degraded gracefully on other browsers.
`text`	Devices typically set off this text from surrounding text, for example, by enclosing it in square brackets, or by underlining it if the device can display bitmap images.

Example

The following example shows how to use the `<anchor>` tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="anchorsample" title="Anhors-Tasks">
        <oui:p>
            Sample Anchors
            <oui:anchor title="tbox1" accesskey="1">
                <oui:go
                    href="http://127.0.0.1/test/timebox1.jsp"/>
                Timebox1
            </oui:anchor>
            <oui:anchor title="tbox2" accesskey="2">
                <oui:go
                    href="http://127.0.0.1/test/timebox2.jsp"/>
                Timebox2
            </oui:anchor>
            <oui:anchor title="Prev" accesskey="3">
                <oui:prev/>
                Previous TBox
            </oui:anchor>
        </oui:p>
    </oui:card>
</oui:wml>
```

<appendix>

Description

The <appendix> tag encapsulates any text or input field that you want to append to a menu. It can only be used within a <oui:combo_menu> tag. See <combo_menu> for more information.

Attributes

Table 1-8. <appendix> tag attributes

Name	Required	rtexprvalue	Description
title	Yes	true	Title of the appendix card and the last menu item on text-based browsers.

Syntax

```
<oui:appendix title="label">
    ...any valid combination of text, <oui:br/>, <oui:p>, and
        <oui:input> tags
</oui:appendix>
```

Example

See <combo_menu> for an example.

<b>

Description

The <b> tag is a direct counterpart of the WML <b> element. It specifies bold text.

Attributes

None

Syntax

```
<oui:b>text</oui:b>
```

In the syntax for this tag, `text` is the text to display in bold type.

<big>

Description

The <big> tag is a direct counterpart of the WML <big> element. It specifies large font text.

Attributes

None

Syntax

```
<oui:big>text</oui:big>
```

In the syntax for this tag, `text` is the text to display in large font.

IMPORTANT Support for this element is available only in version 4.0 or later of the Openwave Mobile Browser.

<body_pager>

Description

The <body_pager> tag breaks up a large block of text and builds an ordered list of concatenated pages. This tag ensures that larger amounts of text are sent to devices that are capable of displaying it. Similarly, smaller amounts of text are sent to devices that are only capable of displaying smaller volumes of text. Using the <body_pager> tag, you can optimize text rendering for any device type.

Attributes

Table 1-9. <body_pager> tag attributes

Name	Required	rtexprvalue	Description
char_set	No	true	Specifies the character set to be used for the WML code.
chunk_size	No	true	Sets the size (in bytes) for each chunk of text. Default is 420 bytes.
title	No	true	Sets the card title.
text_link_forward	No	true	Sets the label for the primary softkey that is used to link to the next page. The default is Next.
text_link_exit	No	true	Sets the label for the secondary softkey used to exit from the <body_pager>. The default is Exit.
url_link_exit	Yes	true	Sets the URL to open when the user exits <body_pager>.
ttl	No	true	Sets the time to live (TTL) or the length of time (in seconds) that a device keeps the deck in cache/memory.

Syntax

```
<oui:body_pager url_link_exit="URL"
            char_set="character set"
            chunk_size="bytes"
            text_link_forward="label"
            text_link_exit="label"
            ttl="seconds">
    content and text
</oui:body_pager>
```

In the syntax for this tag, `content` and `text` are defined as follows.

Table 1-10. <body_pager> tag, content and text definitions

`content`	Any of the following can serve as `content` in the `<body_pager>` tag: • `<oui:bp_header>` • `<oui:bp_chunk>` • `<oui:bp_percentage>` • `<oui:bp_footer>` • `<oui:br>`
`text`	The block of text that you want to split into multiple pages.

Example

The following example shows how to use the <body_pager> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
    <oui:body_pager char_set="iso-8859-1"
                    chunk_size="100"
                    title="Tiger Woods"
                    text_link_forward="Skip"
                    text_link_exit="Done"
                    url_link_exit="http://www.openwave.com/">

    <oui:bp_header>
        <oui:p>Golf News Flash!</oui:p>
    </oui:bp_header>

    LYTHAM St. Annes, England(AP) - Those who wondered whether
    Tiger Woods could stay out of the 196 bunkers that dot the
    links course of Royal Lytham and St. Annes didn't have to
    wait long.

    By the fourth hole Thursday at the British Open, he found
    his first.

    He wound up in five of them during a round of even-par 71
    that left him six strokes behind Colin Montgomerie, his
    largest first-round deficit in four years at the British
    Open.

    And bunkers were the least of his concerns.

    Woods stretched his arm out to the right and yelled "Fore!"
    on three occasions. He failed to birdie any of the par 5s,
    two of them with the wind at his back. The putts that fell
    on command when he was winning four straight majors seemed
    to defy gravity.

    <oui:bp_percentage post_text="%" pre_text="Less" />

    <oui:bp_footer>
        <oui:p>Brought to you by Openwave.</oui:p>
    </oui:bp_footer>

</oui:body_pager>
```

<bp_chunk>

Description

The <bp_chunk> tag adds a chunk to a <body_pager> tag. A chunk is simply a group of text or characters. The chunk size is configurable, but it has a default size of 420 bytes. A page is a collection of one or more chunks. In the context of <body_pager>, a page is synonymous with a WML card. By default, a page is set to contain a maximum of two chunks for Nokia browsers and three chunks for Openwave browsers.

You can use <bp_chunk> to split a long block of text manually instead of allowing the <body_pager> tag to split the text into separate chunks automatically.

Attributes

None

Syntax

```
<oui:bp_chunk>
    ...any valid combination of text and <oui:br/>
</oui:bp_chunk>
```

Example

The following example shows how to use the <bp_chunk> tag.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:body_pager title="Tiger Woods"
                text_link_forward="Skip"
                text_link_exit="Done"
                url_link_exit="http://www.openwave.com" >

    <oui:bp_header>
        <oui:p>Golf News Flash!</oui:p>
    </oui:bp_header>

    <oui:bp_chunk>
        LYTHAM St. Annes, England(AP) - Those who wondered
        whether Tiger Woods could stay out of the 196 bunkers
        that dot the links course of Royal Lytham and St. Annes
        didn't have to wait long. By the fourth hole Thursday
        at the British Open, he found his first.
    </oui:bp_chunk>

    <oui:bp_chunk>
        He wound up in five of them during a round of even-par
        71 that left him six strokes behind Colin Montgomerie,
        his largest first-round deficit in four years at the
        British Open.
    </oui:bp_chunk>

    <oui:bp_chunk>
        And bunkers were the least of his concerns.
    </oui:bp_chunk>

    <oui:bp_chunk>
        Woods stretched his arm out to the right and yelled
        "Fore!" on three occasions. He failed to birdie any of
        the par 5s, two of them with the wind at his back. The
        putts that fell on command when he was winning four
        straight majors seemed to defy gravity.
    </oui:bp_chunk>

    <oui:bp_percentage post_text="%" pre_text="Less" />

    <oui:bp_footer>
        <oui:p>Brought to you by Openwave.</oui:p>
    </oui:bp_footer>
</oui:body_pager>
```

<bp_footer>

Description

The <bp_footer> tag adds a footer to the contents of the text of a <body_pager>. The contents of <bp_footer> appear at the end of every page generated by <body_pager>.

Attributes

None

Syntax

```
<oui:bp_footer>
    ...any valid combination of text, <oui:br>, <oui:img>, and
        <oui:input>
</oui:bp_footer>
```

Example

See <body_pager> for an example.

<bp_header>

Description

The <bp_header> tag adds a header to the text of a <body_pager>. The contents of <bp_header> appear at the beginning of every page generated by <body_pager>.

Attributes

None

Syntax

```
<oui:bp_header>
    ...any valid combination of text, <oui:br>, <oui:img>, and
        <oui:input>
</oui:bp_header>
```

Example

See <body_pager> for an example.

<bp_percentage>

Description

The <bp_percentage> tag adds a percentage value that appears at the end of every page, but before the footer if one is used, indicating how much of the entire text contained in a <body_pager> has been displayed so far.

Attributes

Table 1-11. <bp_percentage> tag attributes

Name	Required	rtexprvalue	Description
pre_text	No	true	Sets the text that appears before the percentage value (for example, less 50).
post_text	No	true	Sets the text that appears after the percentage value (for example, 50% read).

Syntax

```
<oui:bp_percentage pre_text="text" post_text="text" />
```

Example

See <body_pager> for an example.

Description

The <br/> element specifies a line break, that is, it causes the device to display the following text or image on a new line.

Attributes

None

Syntax

```
<oui:br/>
```

<button>

Description

The <button> tag adds a navigation button for the graphical mobile browser that acts independently of all other navigation patterns. You can present an image within a button. On text-based browsers, the button is rendered as a hyperlink. The same is true for the graphical mobile browser when the extensions are disabled.

Use the <button> tag only when alternatives aren't possible. The alternative is to use the <oui:menus> tag and the card paths: <oui:primary_path>, <oui:secondary_path>, <oui:side_path>.

Attributes

Table 1-12. <button> tag attributes

Name	Required	rtexprvalue	Description
href	Yes	true	Sets the URL to open when the user presses the button.
label	Yes	true	Sets the text that appears on the button for the graphical mobile browser or the label representing the link for text-based browsers.
upguipic	No	true	Sets the image that appears in the button instead of the button label for the graphical mobile browser.

Syntax

```
<oui:button href="URL" label="text"
            upguipic="URL of the image" />
```

Example

The following example shows how to use the <button> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="buttonsample" title="Button">
        <oui:p>
            This is a button:
            <oui:button href="www.openwave.com"
                label="OPWV Home" upguipic="opwvlogo.wbmp" />
        </oui:p>
    </oui:card>
</oui:wml>
```

<caller>

Description

The <caller> tag enables users to initiate a telephone call from a list of contacts, an order form, or a phone number query. It uses the make call (MC) function of the Wireless Telephony Application Interface (WTAI). The graphical mobile browser supports the MC function, but the Nokia browser does not. This tag degrades gracefully for browsers that do not support the MC function.

You can use this tag in one of two ways. The first (without specifying the href attribute) allows you to define some of the text that tells users what they are required to do. In this case, OUI builds an extra card automatically and no further action is required on your part. The second (by using the href attribute) lets you define the URL of a card or a deck to which non-WTAI handsets are to be directed. In this case, you must make sure that the card or deck exists. However, because you are creating the card, you can design its appearance.

Attributes

Table 1-13. <caller> tag attributes

Name	Required	rtexprvalue	Description
phone_number	Yes	true	The phone number to call.
text	No	true	Description of the number used by browsers that do not support WTAI.
href	No	true	Specifies the URL of the custom card for browsers that do not support WTAI.

Syntax

```
<oui:caller phone_number="phone number" text="instruction"
    href="URL" />
```

Example

The following example shows how to use the <caller> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="callersample" title="Test Caller">
        <oui:p>
            This is a sample caller
            <oui:primary_path short_label="Call">
                <oui:caller phone_number="4551313612"
                    text="Call Jayce" />
            </oui:primary_path>
        </oui:p>
    </oui:card>
</oui:wml>
```

<card>

Description

The OUI <card> tag encapsulates the WML <card> element. Like its WML counterpart, the OUI <card> tag specifies an interaction between the user and the device.

The OUI <card> tag makes navigation as intuitive as possible, implementing navigation optimally across different browsers. For phones using Openwave browsers, optimal usability is achieved through softkeys; phones from other vendors are better served with hyperlinks. With the OUI <card> tag, you can also require all navigation to be carried out with hyperlinks, giving you the flexibility to choose how you want to implement navigation. For more information on how to enforce certain features, including navigation, card titles, and redefinition of the <prev> task, see the <rendering_directive> tag.

The OUI <card> tag optimizes the user interface on graphical mobile browsers by rendering the primary path (main activity) as a button if there is a widget in the card. Widgets include the <oui:button>, <oui:check>, <oui:input>, <oui:picker>, <oui:popup>, and <oui:radio> tags. This makes it easier for the user to access the widget through the primary softkey without having to share the key with the primary path (main activity) of the card.

Attributes

Table 1-14. <card> tag attributes

Name	Required	rtexprvalue	Description
`id`	No	true	A unique name for the card within the deck. The name acts as a fragment anchor for navigating to that card. For example, you can specify `<go href="#cardname"/>` to navigate to the card.
`title`	No	true	Specifies a brief label for the card. The Openwave text-based browser uses the label as the default bookmark name when the user bookmarks the card. Some devices might use it for other purposes, such as popup tooltips. **NOTE:** Some browsers do not display the title as part of the card content. The phone manufacturer may optionally display the title if there is sufficient screen real estate. You can force the card title to be displayed by using the `<rendering_directive>` tag.
`ontimer`	No	true	Specifies the URL to open if a specified `<timer>` element fires. This attribute represents an abbreviated form of the `<onevent>` element.
`onenterforward`	No	true	Specifies the URL to open if the user navigates to this card through a `<go>` task. This attribute represents an abbreviated form of the `<onevent>` element.

Table 1-14. <card> tag attributes *(continued)*

Name	Required	rtexprvalue	Description
onenterbackward	No	true	Specifies the URL to open if the user navigates to this card through a <prev> task. This attribute represents an abbreviated form of the <onevent> element.
newcontext	No	true	true | false Specifies whether the device should initialize the context whenever the user navigates to the card through a <go> task. Specifying newcontext="true" removes all context-specific variables, clears the history stack, and resets the device state to a well-known value.
ordered	No	true	true | false Specifies the organization of card content. ordered="true" (the default value) causes the device to display content in a fixed sequence. ordered="false" creates an elective form (see "Elective Forms vs. Wizards" in the *Openwave Usability Guidelines*), which displays the content all at once and allows users to select the input field to activate.

Syntax

```
<oui:wml>
    <oui:card id="name"
          title="label"
          newcontext="boolean"
          ordered="true"
          onenterforward="url"
          onenterbackward="url"
          ontimer="url">
        content
    </oui:card>
</oui:wml>
```

In the syntax for this tag, `content` represents the WML card definition and consists of one or more of the following elements.

Table 1-15. <card> tag syntax elements

`content`	You can specify any of the following elements in your card definition: • `<oui:a>` • `<oui:anchor>` • `<oui:combo_menu>` • `<oui:do>` • `<oui:img>` • `<oui:input>` • `<oui:menu>` • `<oui:onevent>` • `<oui:p>` • `<oui:picker>` • `<oui:popup>` • `<oui:primary_path>` • `<oui:secondary_path>` • `<oui:select>` • `<oui:side_path>` • `<oui:table>` • `<oui:task_menu>` • `<oui:timer>` See the corresponding tag definitions in this chapter for more information.

IMPORTANT The contents of a `<card>` element must be in the following order.

- `<oui:onevent>`
- `<oui:timer>`
- `<oui:do>`

With the exception of the elements listed above, devices display these elements in the order in which you specify them.

Examples

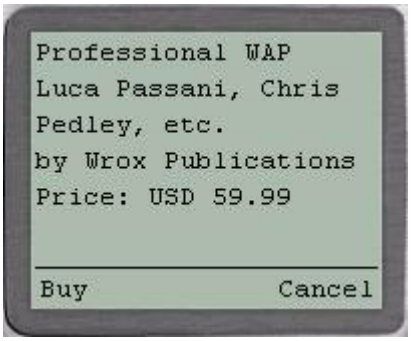
The following three examples show how to use the <card> tag.

Example 1: Defining a primary and secondary path for a card

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="cardsample" title="Borders Bookstore">
        <oui:p>
            Professional WAP <br/>
            Luca Passani, Chris Pedley, et al. <br/>
            by Wrox Publications <br/>
            Price: USD 59.99
        </oui:p>
        <oui:primary_path href="#buy" short_label="Buy" />
        <oui:secondary_path href="#cancel"
            short_label="Cancel" />
    </oui:card>
</oui:wml>
```

The following figure illustrates this example on a text-based browser.

Figure 1-1. <card> tag rendered on a text-based browser



Example 2: Binding a <go> task to an event

The following two code samples illustrate the relationship between the onenterforward attribute and the <onevent> element. In this case, specifying onenterforward as an attribute of the <card> element is exactly the same as specifying the <onevent> element with type="onenterforward". Both cards produce the same result.

```
<%@ taglib uri="/
    WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card onenterforward="url">
        <oui:p>
            Hello World!
        </oui:p>
    </oui:card>
</oui:wml>
```

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card onenterforward="url">
        <oui:onevent
            type="onenterforward">
            <oui:go href="url"/>
        </oui:onevent>
        <oui:p>
            Hello World!
        </oui:p>
    </oui:card>
</oui:wml>
```

Using the abbreviated form implicitly associates a <go> task with the event. The main reasons to use the expanded form are to associate a <prev>, or <refresh> task instead, to encapsulate the setting of variables or postfields, or to include such elements as sendreturner, which cannot be performed with the shortened form. See the <onevent> tag for more information.

Example 3: Ordered option

You can use the `ordered` attribute to specify how the device displays multiple content items within a card.

Setting the `ordered` attribute to `True` displays the items in a linear sequence (the default value). Use this setting for short forms containing mostly required fields. If a device cannot display all the items on one screen, it divides them into multiple screens based on the following:

- Field groupings specified by one or more `<fieldset>` elements, if any. See the `<fieldset>` tag for more information.
- Individual fields, with the text preceding each item definition used as a prompt.

The following example shows how to use the `ordered` attribute.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="orderedsample" title="Login" ordered="false">
        <oui:p>
            Enter Username:
            <oui:input type="text" name="username"
                title="User Name" />
            Enter Password:
            <oui:input type="text" name="password"
                title="Password" />
        </oui:p>
        <oui:primary_path href="#login" short_label="Enter" />
        <oui:secondary_path href="#cancel"
            short_label="Cancel" />
    </oui:card>
</oui:wml>
```

The following figures illustrate card navigation within the deck using the code shown above. Figure 1-2 shows the browser screen when `ordered` is set to `False`. Figure 1-3 and Figure 1-4 show the same application when ordered is set to `True` (the default).

Figure 1-2. Card navigation with the ordered attribute set to False

The input fields are presented all at once. The user can choose which input field to use or skip the first input field.

Figure 1-3. Card navigation with the ordered attribute set to True: Screen 1

This is the first screen. The user enters a username and clicks OK to move to the next input field.

Figure 1-4. Card navigation with the ordered attribute set to True: Screen 2

The input fields are presented in the order defined in the deck. The user is not allowed to skip to the next field without passing through the first input field.

<check>

Description

The <check> tag specifies a checkbox on graphical and Nokia browsers. In text-based browsers, checkboxes are rendered in the form of multiple-selection lists.

Attributes

Table 1-16. <check> tag attributes

Name	Required	rtexprvalue	Description
name	No	true	The name of the variable in which the device stores the values associated with the option(s) chosen by the user. The value associated with each option comes from the <option> element value attribute. The values in the specified variable determine the default selection(s) when the device displays the <check> element. If the variable has no value, the device sets it to the values specified for the default attribute. If you do not specify a default value, the device initializes the variable to an empty string (“”). In the case of multiple selections, the values are stored as a semicolon-separated list.
iname	No	true	Identical to the name attribute except for the following: • The specified variable stores the index values associated with the option(s) chosen by the user. The index value associated with each option comes from its position in the <check> list, starting with 1. If the user has not selected an option, the index value is 0. • The default value is specified by the ivalue attribute.

Table 1-16. <check> tag attributes *(continued)*

Name	Required	rtexprvalue	Description
value	No	true	A string specifying the default values for the variable specified by the name attribute. If the name attribute already has a value when the user navigates to the <check> element, the device ignores the value attribute. If the name attribute does not already have a value, the device sets it to the value specified by the value attribute.
ivalue	No	true	Identical to the value attribute except for the following: • The specified string contains the default index values for the variable specified by the iname attribute.

Syntax

```
<check name="variable" value="default"
    iname="index_var" ivalue="default">
    content
</check>
```

Example

The following example shows how to use the <check> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="checksample" title="Alarms">
        <oui:p>
            <oui:check name="alarm">
                <oui:option value="ocean">Ocean</oui:option>
                <oui:option value="birds">Birds</oui:option>
                <oui:option value="brook">Brook</oui:option>
            </oui:check>
        </oui:p>
    </oui:card>
</oui:wml>
```

<combo_menu>

Description

The <combo_menu> tag combines the <oui:menu> and <oui:appendix> tags. You can use it to build advanced user interfaces on graphical browsers, while ensuring that older browsers can degrade gracefully.

Attributes

None

Syntax

```
<oui:combo_menu>
    <oui:menu>
        menu content
    </oui:menu>
    <oui:appendix>
    appendix content
    </oui:appendix>
</oui:combo_menu>
```

Example

The following example shows how to use the <combo_menu> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="stocks" title="Stock Quotes">
        <oui:combo_menu>
            <oui:menu>
                <oui:menu_item href="get_quote.jsp?symbol=OPWV"
                               title="OPWV"
                               text="OPWV 16.20 0.05-"/>
                <oui:menu_item href="get_quote.jsp?symbol=ACN"
                               title="ACN"
                               text="ACN  15.07 0.22-"/>
                <oui:menu_item href="get_quote.jsp?symbol=CSCO"
                               title="CSCO"
                               text="CSCO 17.13 0.04+"/>
            </oui:menu>
            <oui:appendix title="Other Symbols">
                <oui:p>
                    Get Quotes:
                    <oui:input type="text" name="symbol"
                        value="" title="Stock Symbol" />
                </oui:p>
            </oui:appendix>
        </oui:combo_menu>
    </oui:card>
</oui:wml>
```

<condition>

Description

The <condition> tag specifies a condition that evaluates to true or false. It is especially useful for defining complex conditions using the <conditionlist> tag (see "<conditionlist>" on page 39).

Attributes

Table 1-17. <condition> tag attributes

Name	Required	rtexprvalue	Description
useragent	No	true	A string that identifies the browser being used. This string is compared to the contents of the HTTP request header HTTP_USER_AGENT. The user agent header includes the string "UP.Browser" for Openwave browsers using Openwave gateways, while it includes the string "Nokia" for Nokia browsers. In most cases, Openwave browsers using non-Openwave gateways include the string "UP/".
agentfamily	No	true	UPText | UPGUI | Nokia | MSIE5 | Generic Identifies the family of the browser being used.
gateway_vendor	No	true	Openwave | Nokia | CMG Identifies the vendor of the gateway.
negate	No	true	true | false Negates the evaluation of the condition's useragent, agentfamily, gateway_vendor, and version.
version	No	true	Identifies the browser version.

Syntax

```
<oui:condition useragent="browser"
               agentfamily="family of browser"
               gateway_vendor="vendor"
               version="browser version" />
```

Example

See <conditionlist> for an example.

<conditionlist>

Description

The <conditionlist> tag groups and evaluates one or more <oui:condition> tags.

Attributes

Table 1-18. <conditionlist> tag attributes

Name	Required	rtexprvalue	Description
logic	No	true	and | or Defines the logic to be applied to all of the conditions. Unless otherwise specified, the default logic is and.
negate	No	true	true | false Negates the resulting evaluation of the group of conditions.

Syntax

```
<oui:if>
    <oui:conditionlist logic="and/or">
        conditions
    </oui:conditionlist>
    other conditional tags
</oui:if>
```

In the syntax for this tag, you can specify *conditions* using:

- <oui:condition/>
- <oui:conditionlist/> (nested <oui:conditionlist/>)

For *other conditional tags*, you must specify the code to be rendered when the condition is satisfied. Alternatively, you can specify code to be rendered when the condition is not satisfied:

- <oui:then>
- <oui:else>

Example

The following example shows how to use the <conditionlist> tag to distinguish an "all-openwave" handset from other handsets.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="condtag" title="Conditional Tags">
        <oui:p align="left">
            Openwave Detector activated...
            <oui:if>
                <oui:conditionlist logic="AND">
                    <oui:condition gateway_vendor="openwave" />
                    <oui:conditionlist logic="OR">
                        <oui:condition agentfamily="upgui" />
                        <oui:condition agentfamily="uptext" />
                    </oui:conditionlist>
                </oui:conditionlist>
                <oui:then>
                    You are using an Openwave browser and
                    you are connecting through an Openwave
                    gateway.
                </oui:then>
                <oui:else>
                    Your browser, your gateway, or both were
                    are not Openwave.
                </oui:else>
            </oui:if>
        </oui:p>
    </oui:card>
</oui:wml>
```

<do>

Description

The <do> tag is a direct counterpart of the WML <do> element. It can be attached to cards and templates and effectively implements navigational elements on some browsers, typically the Openwave Mobile Browser.

Attributes

Table 1-19. <do> tag attributes

Name	Required	rtexprvalue	Description
type	Yes	true	Required. Identifies the generic user interface mechanism that triggers the specified <do> element task. The mobile browser software does not currently support this attribute for the following TYPE values: type="delete" type="help" type="prev"
label	No	true	Specifies a text string suitable for labeling the user widget. For example, if you bind a task to the ACCEPT key, the device displays this value as the function key label. If you do not specify the label attribute, the device uses the word OK as the default ACCEPT key label. To ensure compatibility on a wide range of devices, the label should be a maximum of five characters. Devices that do not support dynamic labeling ignore the label attribute.
name	No	true	Specifies a name for the <do> element. If a card-level <do> element (for example, defined within a <card> element) has the same name as a deck-level <do> element (for example, defined within a <template> element), the card-level binding overrides the deck-level binding.
optional	No	true	true | false Specifies whether the device can ignore this element. Default is false.

You can specify the following values for the type attribute. All types are reserved except where noted.:

Table 1-20. <do> tag attribute values

Value	Action
accept	Invokes the ACCEPT mechanism (function key, button, and so on). To optimize usability across browsers, use the <primary_path> tag instead of <do type="accept">.
delete	Invokes the DELETE mechanism (function key, button, and so on).
help	Invokes the HELP mechanism (may be context sensitive).
options	Invokes the OPTIONS mechanism (function key, button, and so on). To optimize usability across browsers, use the <secondary_path> or <side_path> tags instead of <do type="options">.
prev	Navigates to a card by invoking the PREV mechanism.
reset	Invokes the RESET mechanism (clears or resets the current device state). The mobile browser does not currently support this value.
unknown	Invokes an unknown mechanism (equivalent to TYPE=""). The mobile browser does not currently support this value.
vnd.co-type	Invokes a vendor-specific mechanism where co identifies the vendor and type identifies the action (not reserved).
X-*, x-*	Future use (not reserved). The mobile browser does not currently support this value.

None of these type values imply a specific user interface mechanism. Some devices map each type to a physical key, while others map them to context-dependent gestures (for example, pressing or press-holding a jog shuttle). When designing your user interface, keep in mind that you can neither specify nor assume the particular mechanism that a device uses.

NOTE If you define multiple <do> elements of the same type in one card, you should specify the name attribute for each <do> element to uniquely identify each instance of the same type. When defining multiple <do> elements, the OPTIONS softkey is labeled Menu. When the user selects the OPTIONS key, a pop-up menu appears above the OPTIONS key. The label for each <do type="options"> is rendered in the order in which they are placed in the WML card. Users can select an option either by pressing the key accelerator associated with the item in the pop-up menu or by using the up/down navigation to highlight one of the options and then pressing the ACCEPT key.

Syntax

```
<oui:do type="type" label="label" >
    task
</oui:do>
```

In the syntax for this tag, `task` represents the action to perform when the user activates the specified interface mechanism.

Table 1-21. <do> tag, task description

`task`	You must bind one of the following `task` elements to the user interface mechanism: • `<oui:go>` • `<oui:prev>` • `<oui:noop>` • `<oui:refresh>` • `<oui:caller>` You can also anchor the Openwave extension tasks `<spawn>`, `<exit>`, and `<throw>` using the `<raw_mode>` tag. These tasks are not directly supported because they cannot be degraded gracefully on other browsers. You can also anchor an `<img>` tag within the `<do>` tag using the `<raw_mode>` tag. This is not directly supported because usability is not optimized with pictures on softkeys.

<else>

Description

The <else> tag specifies a branching of code based on the evaluation of a condition or a set of conditions. It is synonymous with the else keyword in most programming languages. If the conditions evaluate to False, the content of the <else> tag is rendered. This tag can only be used inside an <oui:if> tag.

Syntax

```
<oui:if>
    conditions
    <oui:then>
        any valid combination of OUI tags
    <oui:then>
    <oui:else>
        any valid combination of OUI tags
    </oui:else>
</oui:if>
```

In the syntax for this tag, conditions are specified by the <oui:condition> or <oui:conditionlist> tag.

Example

See the <conditionlist> tag for an example.

Description

The tag is a direct counterpart of the WML element. It specifies emphasized text.

Attributes

None

Syntax

```
<oui:em>text</oui:em>
```

In the syntax for this tag, `text` is the text to display in emphasized font.

<fieldset>

Description

The <fieldset> tag is a direct counterpart of the WML <fieldset> element. You can use it to group multiple text or input items within a card. By specifying one or more <fieldset> elements, you can control how the device presents card content in order to simplify user navigation.

Attributes

Table 1-22. <fieldset> tag attributes

Name	Required	rtexprvalue	Description
title	No	true	Specifies a brief label for the <fieldset> group. Some devices use the label as a title when displaying the <fieldset> content. Others use it as a label for a user interface mechanism that lets the user navigate to the <fieldset> content. For example, if a device cannot display all card content on one screen and ordered="true", the Mobile Browser uses the title to identify this group of items on a summary-level menu.

Syntax

```
<oui:fieldset title="label">content</oui:fieldset>
```

In the syntax for this tag, content represents the items to group together and consists of one or more of the following elements.

Table 1-23. <fieldset> tag, content description

content	You can specify any of the following elements in a <fieldset> group: • <oui:fieldset> (a nested <oui:fieldset>) • <oui:input> • <oui:select>
text	You can also specify formatted text in a <fieldset> group. The device uses this text in different ways, depending on the elements you specify—for example, to prompt the user for input or to describe various options.

Devices display these elements in the order in which you specify them.

<form>

Description

At times, applications require users to insert several pieces of information. In these cases, you should use the <form> tag. The <form> tag contains a variety of selection lists (checkbox, pop-up menu, radio buttons), input fields, text, and buttons that are used to simplify and guide data entry. With the <form> tag, you can rely on the rendering strategy of OUI to optimize the usability of applications across various browsers.

Attributes

Table 1-24. <form> tag attributes

Name	Required	rtexprvalue	Description
id	No	true	A unique name for the form in the deck. The name acts as a fragment anchor for navigating to that form. For example, you can specify <go href="#formname"/> to navigate to the form.
title	No	true	Specifies a brief label for the card. The Openwave text-based browser uses the label as the default bookmark name when the user bookmarks the card. Some devices use it for other purposes, such as pop-up tooltips. **NOTE:** Some browsers do not display the title as part of the card content. The phone manufacturer may display the title if there is sufficient screen real estate. However, you can force the card title to be displayed by using the <oui:rendering_directive> tag.
ontimer	No	true	Sets the URL to open when the timer expires.

Table 1-24. <form> tag attributes *(continued)*

Name	Required	rtexprvalue	Description
onenterforward	No	true	Sets the URL to open when the user navigates to the form in a forward direction. This is a direct counterpart of the WML onenterforward attribute for the <card> element.
onenterbackward	No	true	Sets the URL to open when the user navigates to the form in a backward direction. This is a direct counterpart of the WML onenterbackward attribute for the <card> element.
newcontext	No	true	Specifies whether the device should initialize the context whenever the user navigates to the form through a <go> task. This is a direct counterpart of the WML newcontext attribute for the <card> element. Specifying newcontext="true" removes all context-specific variables, clears the history stack, and resets the device state to a well-known value.

Syntax

```
<oui:wml>
    <oui:form id="name"
              title="label"
              newcontext="boolean"
              ordered="true"
              onenterforward="url"
              onenterbackward="url"
              ontimer="url">
              content
    </oui:form>
</oui:wml>
```

In the syntax for this tag, `content` represents the WML card definition and consists of one or more of the following elements:

Table 1-25. <form> tag, content description

`content`	You can specify any of the following elements in your card definition: • `<oui:a>` • `<oui:anchor>` • `<oui:combo_menu>` • `<oui:img>` • `<oui:input>` • `<oui:menu>` • `<oui:p>` • `<oui:picker>` • `<oui:popup>` • `<oui:primary_path>` • `<oui:secondary_path>` • `<oui:select>` • `<oui:side_path>` • `<oui:table>` • `<oui:task_menu>`

Example

The following example shows how to use the `<form>` tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:form id="formsample" title="Login">
        Enter Username:
        <oui:input type="text" name="username"
            title="User Name:" />
        Enter Password:
        <oui:input type="password" name="password"
            title="Password:" />
        <oui:primary_path href="#login" short_label="Enter" />
        <oui:secondary_path href="#cancel"
            short_label="Cancel" />
    </oui:form>
</oui:wml>
```

<go>

Description

The <go> tag is the direct counterpart of the WML <go> element. It is a task element that instructs the device to open a specified URL. If the URL specifies a particular card, the device displays that card. If the URL specifies a deck, the device displays the first card in that deck.

Attributes

Table 1-26. <go> tag attributes

Name	Required	rtexprvalue	Description
`href`	Yes	true	Specifies the destination URL.
`method`	No	true	`get | post` Specifies the HTTP submission method. Specifying `method="POST"` causes the Openwave Mobile Access Gateway to transcode variable data to the character set specified by the HTTP headers defined in your application. You should perform this transcoding if non-ASCII characters (specifically UTF-8) may exist in the data being passed. For more information on character sets and HTTP headers, see the *WML Developer's Guide*. If you do not specify the method attribute but do specify the `postfield` nested element, the device automatically uses the `get` method.
`sendreferer`	No	true	`true | false` Specifies whether the device should include the deck URL in the URL request. Specifying `sendreferer="true"` causes the device to set the `HTTP_REFERER` header to the relative URL of the requesting deck. If you want to restrict access to trusted services, decks that request specified URLs must set this option to `true`.

Table 1-26. <go> tag attributes *(continued)*

Name	Required	rtexprvalue	Description
`accept-charset`	No	true	Specifies the character encodings that your application can handle. The device uses this attribute to transcode data specified by the postfield element. The Openwave Mobile Access Gateway assumes UTF-8 as the default encoding (of which US-ASCII is a subset), so WML services in the United States, Canada, or Australia do not need to use this attribute. You can also omit this attribute if you specify your character sets in the HTTP response header. Note that the `accept-charset` attribute overrides any character encodings you specify in the HTTP header. The syntax for this attribute is a comma- or space-delimited list of IANA character sets; for example, `accept-charset="UTF-8, US-ASCII, ISO-8859-1"`. • For a list of Openwave-supported encoding names, see the *WML Developer's Guide*. • To view the complete IANA character set registry, go to `http://www.iana.org/`.

Syntax

```
<oui:go href="url"
        sendreferer="boolean"
        method="method"
        accept-charset="charset">
        content
</oui:go>
```

In the syntax for this tag, `content` represents the variables to set when opening the specified URL.

Table 1-27. <go> tag, content description

`content`	You can optionally specify one or more variables in a `<go>` statement: • `<oui:postfield>` • `<oui:setvar>`

IMPORTANT Unlike other WML elements that have content, specifying content for the `<go>` element is optional. If you do not specify any content, you must use the syntax `<go attributes/>` rather than `<go attributes>content</go>`.

<head>

Description

The <head> tag is a direct counterpart of the WML <head> element. It specifies information about the deck as a whole, including metadata and access control information. This object is used to exploit advanced functionality of the Openwave platform such as prefetch, bookmark control, and deck caching (time to live), without breaking the code for other browsers.

Syntax

```
<oui:head>
    content
</oui:head>
```

In the syntax for this tag, content represents deck-level header information.

Table 1-28. <head> tag, content description

content	You can optionally specify either of the following elements in a WML deck header: • <oui:access> One only • <oui:meta> One or more

Example

The following example shows how to use the <head> tag to specify that the WML deck should not be cached.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:head>
        <oui:meta forua="true" http_equiv="Cache-Control"
            content="no-cache, must-revalidate" />
    </oui:head>
    <oui:card id="testcard" title="Meta Test" >
        ...
    </oui:card>
</oui:wml>
```

<hr>

Description

The <hr> tag is the direct counterpart of the <hr> WML Openwave extension. This tag creates a user-friendly layout on the card in the graphical mobile browser without breaking the code for other browsers. The default behavior causes a horizontal rule to be drawn across the full width of the phone display. If the length of the line exceeds the dimensions of the display, the line is truncated at the edge of the screen.

Attributes

Table 1-29. <hr> tag attributes

Name	Required	rtexprvalue	Description
size	No	true	Controls the height of the line to be drawn in pixels and has an implied value of 1. Acceptable values are integers.
width	No	true	Controls how many pixels long (horizontally) the line should be drawn. Values can be either an absolute number of pixels or a percentage of the screen width. If the value of the width attribute exceeds the dimensions of the screen, the line is truncated at the screen edge.

Syntax

```
<oui:hr size="pixels" width="pixels or percentage"/>
```

Example

The following example shows how to use the <hr> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="hrsample" title="Employee Info">
        <oui:p>
            Full Name: Alan Lampa
            <oui:br/>
            <oui:hr size="5" width="180"/>
            Primary Contact Info
            <oui:br/>
            Office:+1(612)2770917
            <oui:br/>
            Mobile:+1(612)3107043
            <oui:br/>
            <oui:hr size="5" width="100%"/>
            Personal Info
            <oui:br/>
            Home: +63 2 697-2573
            <oui:br/>
            Yahoo ID: alan_pogi
        </oui:p>
    </oui:card>
</oui:wml>
```

<if>

Description

The <if> tag is a conditional tag that specifies that a block of code is rendered only if certain conditions are satisfied. This tag is synonymous to the if keyword in most programming languages.

Attributes

Table 1-30. <if> tag attributes

Name	Required	rtexprvalue	Description
useragent	No	true	A string to identify the browser being used. This string is compared to the contents of the HTTP request header HTTP_USER_AGENT. The user agent header includes the string "UP.Browser" for Openwave browsers using Openwave gateways, while it includes the string "Nokia" for Nokia browsers. In most cases, Openwave browsers using non-Openwave gateways include the string "UP/".
agentfamily	No	true	UPText | UPGUI | Nokia | MSIE5 | Generic Identifies the family of the browser being used.
agentsubfamily	No	true	UPGUI | UPText | Alcatel | Nokia | Nokia 9110 | Generic | MSIE5 Identifies a more specific family of browser being used.
gateway_vendor	No	true	Openwave | Nokia | CMG Identifies the vendor of the gateway.

Table 1-30. <if> tag attributes *(continued)*

Name	Required	rtexprvalue	Description
`version`	No	true	Identifies the browser version.
`special`	No	true	`allopenwave | allnokia` Specifies whether the handset is using an Openwave browser with an Openwave gateway or a Nokia browser with a Nokia gateway. This is a short version for specifying a list of conditions checking for the browser-gateway combination.
`negate`	No	true	Negates the evaluation of the conditions: `useragent`, `agentfamily`, `agentsubfamily`, `gateway_vendor`, `version`, and `special`.

Syntax

Simple syntax

```
<oui:if [simplecondition="value"]>
    content
</oui:if>
```

In the syntax for this tag, `simplecondition` and `content` are defined as follows.

Table 1-31. <if> tag, simplecondition and content descriptions

`simplecondition`	`useragent ="useragent or browser"`
	`agentfamily ="family of browsers"`
	`agentsubfamily =”subfamily of browser”`
	`gateway_vendor ="vendor"`
	`version ="browser version"`

`content`	A valid combination of text and OUI tags.

Complex syntax

```
<oui:if>
    condition or condition list
    <oui:then>
        content
    </oui:then>
    <oui:else>
        content
    </oui:else>
</oui:if>
```

In the syntax for this tag, `simplecondition` and `content` are defined as follows.

Table 1-32. <if> tag, condition and condition list description

`condition`	`<oui:condition [simplecondition="value"]>`

`condition list`	`<oui:condition_list logic="logic operator">`
	`condition or condition list`
	`<oui:condition_list>`

NOTE The `logic` attribute can be `OR`, `AND`, or `NOT`. If you omit the `logic` attribute, `AND` logic is the default condition evaluation.

Example

See the `<conditionlist>` tag for an example.

Description

The <img> tag encapsulates the WML <img> element so that you can specify different image types for different classes of devices. This means that you can produce multiple versions of an image and serve the image in the appropriate type to the requesting device. For example, you can serve an image in WBMP format to Nokia and text-based browsers, while serving the same image in PNG format to the graphical Openwave Mobile Browser.

NOTE Not all graphical Openwave Mobile Browser phones support PNG format images.

Attributes

Table 1-33. <img> tag attributes

Name	Required	rtexprvalue	Description
alt	Yes	true	Specifies the text to display if the device does not support images or cannot find the specified image.
src	Yes	true	Specifies the URL of the image to be shown. If you specify a valid icon for the localsrc attribute, the device ignores this attribute.
localsrc	No	true	Specifies the name of a known icon to be displayed instead of the src attribute. If the device cannot find the icon in ROM, it attempts to retrieve it from the WAP gateway. If you specify a valid icon (see Figure for a list of icon names), the device ignores the src and alttext attributes, even though they are still required.
align	No	true	top | middle | bottom Specifies the alignment of the image relative to the current line of text.
height	No	true	Specifies the height of the image. If specified as a percentage value, the size is based on the available vertical space and not on the natural size of the image.

Table 1-33. <img> tag attributes *(continued)*

Name	Required	rtexprvalue	Description
`width`	No	true	Specifies the width of the image. If specified as a percentage value, the size is based on the available horizontal space and not on the natural size of the image.
`vspace`	No	true	Specifies the amount of white space to be inserted above and below the image. If specified as a percentage value, the space is based on the available vertical space and not on the natural size of the image.
`hspace`	No	true	Specifies the amount of space to the left and right of the image. The default is zero.

Syntax

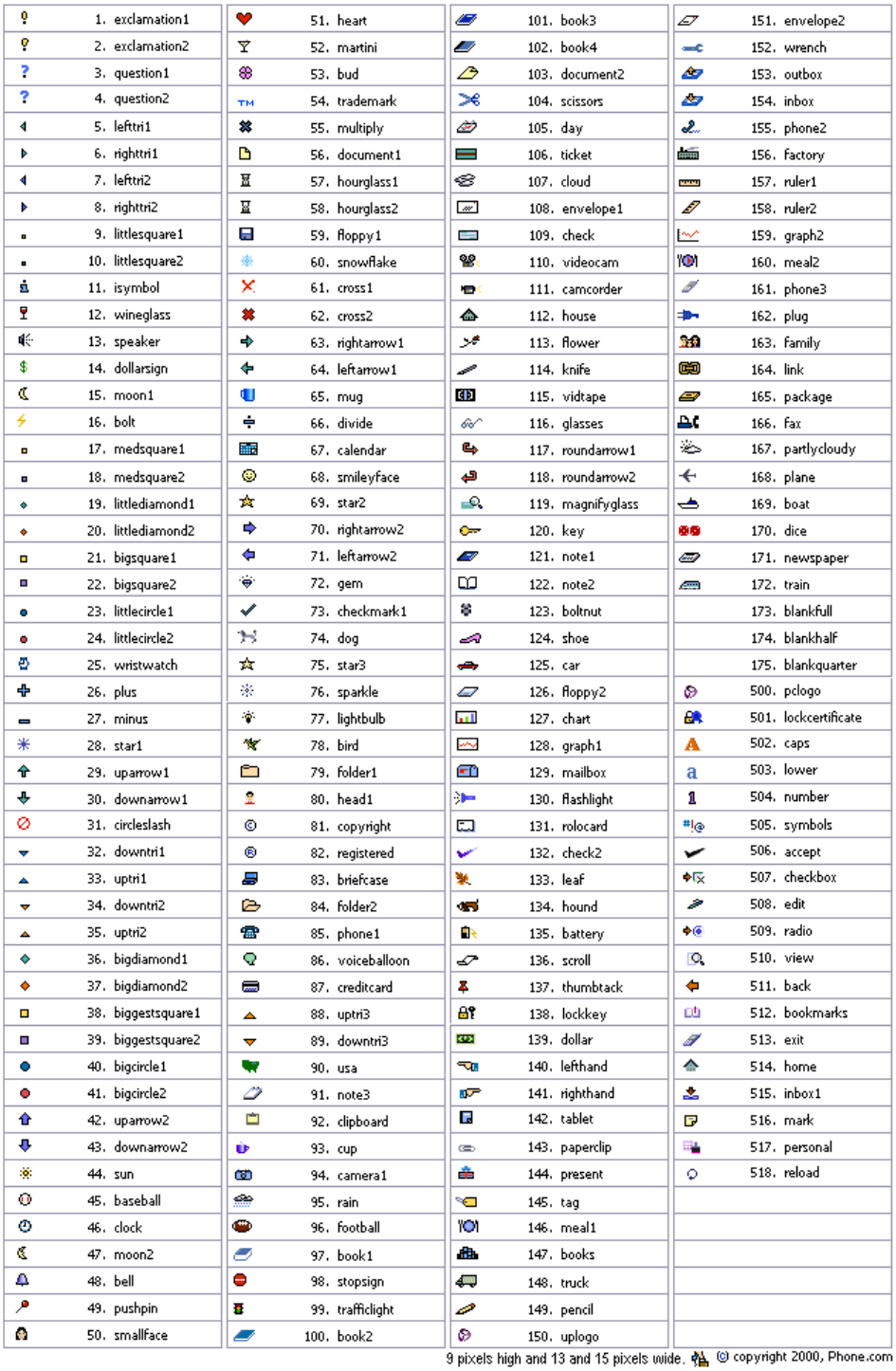
```
<oui:img alt="text"
         src="url"
         localsrc="icon"
         align="alignment"
         height="n"
         width="n"
         vspace="n"
         hspace="n">
        content
</oui:img>
```

In the syntax for this tag, `content` specifies an alternative image to be displayed:

Table 1-34. <img> tag, content description

`content`	You can use the `<oui:additional_src>` tag to specify one or more alternative images to be displayed for a particular browser, device, family of browsers, or family of devices.

Figure 1-5. Predefined Openwave icons



Example

The following example shows how to use the <img> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="card1" title="IMG Test" >
        <oui:p align="center" mode="wrap">
            This tests the img tag.
            Here is the image:
            <oui:img alt="UK Compass"
                src="http://www.openwave.com/
                images/default_logo.wbmp">
            <oui:additional_src agentfamily="UPGUI"
                src="http://www.openwave.com/
                images/opwv.png" />
            <oui:additional_src agentfamily="UPText"
                src="http://www.openwave.com/
                images/opwv.wbmp"/>
            </oui:img>
        </oui:p>
    </oui:card>
</oui:wml>
```

<input>

Description

The <input> tag encapsulates the WML <input> element. It allows the user to enter text that the device assigns to a specified variable. In addition, it minimizes confusion about the input format by allowing developers to specify different input formats (or masks) for a particular device or family of devices to ensure that adequate feedback is given to the user.

Attributes

Table 1-35. <input> tag attributes

Name	Required	rtexprvalue	Description
type	No	true	text | password Specifies how the device should display text the user enters. Specifying type="text" causes the text to be visible. Specifying type="password" causes the text to be masked (for example, replaced by * characters). The password mode is not encrypted, so you should not rely on it for securing critical data.
name	Yes	true	The name of the variable in which the device stores the text entered by the user. When the device displays the <input> element, the value in the specified variable appears in the entry field.
value	No	true	Specifies the value of the variable named in the name attribute. When the element is displayed and the variable named in the name attribute is not set, the name variable is assigned the value specified in the value attribute. If the name variable already contains a value, the value attribute is ignored. If the value attribute specifies a value that does not conform to the input mask specified by the format attribute, the user agent ignores the value attribute.

Table 1-35. <input> tag attributes *(continued)*

Name	Required	rtexprvalue	Description
`title`	No	true	Specifies a brief label for the input item. Some devices use the label as a tool tip when displaying the input field. Others use it as a label for a user interface mechanism that lets the user navigate to the item. For example, if a device cannot display all card content on one screen and `ordered="true"`, the browser uses the title to identify this input item on a summary-level menu.
`format`	No	true	Specifies a data format that the user entry must match (see "Specifying a Format Mask" on page 65). If you omit this attribute, the device assumes *M (default uppercase first character followed by up to `maxlength` number of mixed-case alphabetic and numeric characters).
`emptytok`	No	true	`true | false` Specifies whether the user can leave the field blank. Specifying `emptyok="true"` indicates that the field is optional. If the user enters a value, however, the device applies any entry requirements that you specify for the format attribute.
`size`	No	true	Controls the on-screen size of the input field. If the `size` attribute is not included, the field grows to accommodate all of the characters that the user enters. If a value is given for the `size` attribute, when the user enters more characters than can be displayed in the field, the characters scroll off the screen to the left. When the user navigates off of the input element, only the beginning characters of the input are visible in the field.
`maxlength`	No	true	Specifies the maximum number of characters the user can enter. If you do not specify the `maxlength` attribute, the mobile browser imposes a limit of 256 characters.

Syntax

```
text
<oui:input name="variable"
           title="label"
           type="type"
           value="value"
           format="specifier"
           emptyok="boolean"
           size="n"
           maxlength="n">
    content
</oui:input>
```

In the syntax for this tag, `content` and `text` represent the following.

Table 1-36. <input> tag, text and content descriptions

`text`	Represents the text and/or image the device displays to prompt the user for entry.
`content`	You can use the `<oui:additional_format>` tag to specify one or more additional formats for a particular device, browser, family of devices, or family of browsers.

Specifying a Format Mask

You can specify the following values for the format attribute.

Table 1-37. <input> tag format masks

Tag	Description
A	Any symbolic or uppercase alphabetic character (no numbers).
a	Any symbolic or lowercase alphabetic character (no numbers).
N	Any numeric character (no symbols or alphabetic characters).
X	Any symbolic, numeric, or uppercase alphabetic character (cannot be changed to lowercase).
x	Any symbolic, numeric, or lowercase alphabetic character (cannot be changed to uppercase).
M	Any symbolic, numeric, or uppercase alphabetic character (can be changed to lowercase). For multiple character input, defaults to uppercase first character.
m	Any symbolic, numeric, or lowercase alphabetic character (can be changed to uppercase). For multiple character input, defaults to lowercase first character.

- To limit the number of characters that users can enter, specify a single digit number before the character tag. For example, `format="3X"` lets the user enter a maximum of three symbolic, numeric, or uppercase alphabetic characters.
- To let users enter an unlimited number of characters, specify an asterisk (*) before the character tag. For example, `format="*a"` lets the user enter any number of symbolic or lowercase alphabetic characters.

Example

The following example shows how to use the <input> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="formatsample" title="Wash. D.C. Phone">
        <oui:p>
            Phone no:
            <oui:input type="text" name="phoneno" value=""
                format="AAAAAAA">
            <oui:additional_format agentfamily="UPText"
                format="\\2\\0\\2\\-NNNNNNN" />
            <oui:additional_format agentfamily="UPGUI"
                format="\\(\\2\\0\\2\\)NNN\\-NNNN" />
            <oui:additional_format agentfamily="Nokia"
                format="NNNNNNN" />
            <oui:additional_format agentfamily="MSIE"
                format="NNN\\XNNNNNNN" />
            </oui:input>
        </oui:p>
        <oui:primary_path href="#process" short_label="OK" />
        <oui:secondary_path href="#exit"
            short_label="Cancel" />
    </oui:card>
</oui:wml>
```

<menu>

Description

The <menu> tag implements a menu as hyperlinks on Nokia phones and as a numbered list on Openwave text-based browsers.

Attributes

None

Syntax

```
<oui:menu>
    content
</oui:menu>
```

In the syntax for this tag, content specifies the menu items (see "<menu_item>" on page 68).

Example

See the <menu_item> tag for an example.

<menu_item>

Description

The <menu_item> tag specifies a particular choice for a selection denoted by the <menu> or <picker_card> tag.

Attributes

Table 1-38. <menu_item> tag attributes

Name	Required	rtexprvalue	Description
title	No	true	A label that identifies the option. The Openwave Mobile Browser uses the title as the ACCEPT key label when the user selects the option. To ensure compatibility on a wide range of devices, label should be a maximum of five characters.
icon	No	true	Specifies the name of a known icon to be displayed to the left of the option. If the device cannot find the icon in ROM, it attempts to retrieve it from the WAP gateway. (see Figure on page 60 for a list of icon names).
href	No	true	Specifies the URL to open when the user selects the option.
text	No	true	The device displays this text to represent the selection item.
value	No	true	Specifies the value to assign to the variable defined in the <oui:picker_card> element var_name attribute when the user selects the option. If you specify a variable reference, the device evaluates the reference before setting the name variable. This attribute is applicable only when the <menu_item> tag is enclosed in a <oui:picker_card> tag. It is ignored if the <menu_item> tag is associated with a <oui:menu> tag.

Syntax

```
<oui:menu>
    <oui:menu_item title="label"
                   href="url"
                   icon="icon"
                   text="text">
        content
    </oui:menu_item>
    ...
</oui:menu>
or
<oui:picker_card>
    <oui:menu_item title="label"
                   href="url"
                   icon="icon"
                   text="text"
                   value="value">
        content
    </oui:menu_item>
    ...
</oui:picker_card>
or
<oui:task_menu>
    <oui:menu_item title="label"
                   href="url"
                   icon="icon"
                   text="text">
        content
    </oui:menu_item>
    ...
</oui:task_menu>
```

In the syntax for this tag, `content` represents an optional task.

Table 1-39. <menu_item> tag, content description

content	• <oui:caller> • <oui:go> • <oui:prev> • <oui:noop> • <oui:refresh>

Example

The following example shows how to use the <menu_item> tag.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="menusample" title="Calendar App">
        <oui:p>
            <oui:menu>
                <oui:menu_item href="sample_task.jsp"
                    title="Today" text="Today (July 11)" />
                <oui:menu_item href="#this_week_view"
                    title="Week" text="Week (9-13)" />
                <oui:menu_item href="#month_view"
                    title="Month" text="Month (July)" />
                <oui:menu_item href="sample_task.jsp"
                    title="Tomorrow"
                    text="Tomorrow (July 12)" />
                <oui:menu_item href="sample_task.jsp"
                    title="Friday" text="Friday (July 13)" />
                <oui:menu_item href="sample_task.jsp"
                    title="Next" text="Next Appoint"
                    icon="clock"/>
                <oui:menu_item href="search_date.jsp"
                    title="Search" text="Pick a date..." />
            </oui:menu>
        </oui:p>
    </oui:card>
</oui:wml>
```

<meta>

Description

The <meta> tag adds the WML <meta> element and is used to specify general information about the deck.

Attributes

Table 1-40. <meta> tag attributes

Name	Required	rtexprvalue	Description
http_equiv	No	true	Can be used in place of name. Indicates that the property should be interpreted as an HTTP header.
name	No	true	Specifies the property name. The browser must ignore any metadata named with this attribute. Servers should not emit WML content containing metadata named with this attribute.
forua	No	true	Specifies that the author intended the property to reach the user agent. If forua="false", an intermediate agent must remove the <meta> element before the document is sent to the client. If the value is true, the metadata of the element must be delivered to the user agent. The method of delivery may vary.
content	No	true	Specifies the metadata value associated with the property attribute.
schema	No	true	Specifies a form or structure that can be used to interpret the property value. Scheme values vary depending on the type of metadata.

Syntax

```
<oui:head>
    <oui:meta http-equiv="Cache-Control"
              content="max-age=time"
              forua= true />
    ...
<oui:/head>
```

Example

The following example shows how to use the <meta> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:head>
        <oui:meta forua="true" http_equiv="Cache-Control"
            content="max-age=600" />
    </oui:head>
    <oui:card id="testcard" title="Meta Test" >
    <oui:p>
        This card will be gone in 60 seconds.
        </oui:p>
    </oui:card>
</oui:wml>
```

<method>

Description

The <method> tag specifies a particular rendering strategy to be enforced on a particular context (for example, <oui:card>, <oui:body_pager>, <oui:menu>, or <oui:form>) specified by <oui:rendering_directive>.

Attributes

Table 1-41. <method> tag attributes

Name	Required	rtexprvalue	Description
name	Yes	true	Specifies the feature to be enforced on the context. See <rendering_directive> for the valid values.

Example

See <rendering_directive> for an example.

<noop>

Description

The <noop> tag is the direct counterpart of the WML <noop> tag.

Syntax

```
<oui:noop/>
```

<onevent>

Description

The <onevent> tag is the direct counterpart of the WML <onevent> element.

Attributes

Table 1-42. <onevent> tag attributes

Name	Required	rtexprvalue	Description
type	Yes	true	Required. Identifies the intrinsic event that triggers the specified <onevent> task. If a card-level <onevent> element (one that is defined in a <oui:card> element) has the same type as a deck-level <onevent> element (one that is defined in a <oui:template> element), the card-level binding overrides the deck-level binding.

Syntax

```
<oui:onevent type="type">task</oui:onevent>
```

In the syntax for this tag, task represents the action to perform when the intrinsic event occurs.

Table 1-43. <onevent> tag, task description

task	You can specify any one of the following actions for the <onevent> element: • <oui:go> • <oui:prev> • <oui:refresh> • <oui:caller> You can also anchor the Openwave extension tasks <spawn>, <exit>, and <throw> using the <oui:raw_mode> tag. These tasks are not directly supported because they cannot be degraded gracefully on other browsers.

You can specify the following values for the `type` attribute.

Table 1-44. <option> tag, type attribute values

Type value	Perform task if…
`onpick`	User selects or deselects an `<option>` item
`onenterforward`	User navigates to a card through a `<go>` task
`onenterbackward`	User navigates to a card through a `<prev>` task or invokes the PREV mechanism (for example, presses the BACK key)
`ontimer`	A specified `<timer>` element expires

<optgroup>

Description

The <optgroup> tag is a direct counterpart of the WML <optgroup> element. You can use it to group multiple <oui:option> or nested <oui:optgroup> elements within a card. By creating option groups, you can specify control information about how the device should present the card content.

Attributes

Table 1-45. <optgroup> tag attributes

Name	Required	rtexprvalue	Description
title	No	true	Specifies a brief label for the <optgroup> group. Some devices use the label as a title when displaying the <optgroup> content. Others use it as a label for a user interface mechanism that lets the user navigate to the <optgroup> content.

Syntax

```
<oui:optgroup title="label">content</oui:optgroup>
```

In the syntax for this tag, content represents one or more of the following.

Table 1-46. <optgroup> tag, content description

content	You can specify any of the following elements: • <oui:optgroup> (a nested <oui:optgroup> element) • <oui:option>

Devices display these elements in the order in which you specify them.

<option>

Description

The <option> tag is the direct counterpart of the WML <option> element.

Attributes

Table 1-47. <option> tag attributes

Name	Required	rtexprvalue	Description
value	No	true	Specifies the value to assign to the variable defined in the name attribute of the <oui:check>, <oui:popup>, or <oui:radio> element when the user selects the option. If you specify a variable reference, the device evaluates the reference before setting the name variable.

Syntax

```
<oui:option value="value">
    content
</oui:option>
```

In the syntax for this tag, content represents the text that the device displays to represent the particular selection. You can also place an image in the <option> tags using the <oui:raw_mode> tag.

Example

See the descriptions of <check>, <picker>, <popup>, and <radio> for examples.

<p>

Description

The <p> tag is the direct counterpart of the WML <p> element.

Attributes

Table 1-48. <p> tag attributes

Name	Required	rtexprvalue	Description
`align`	No	true	`left | right | center` Specifies line alignment relative to the display area. Specifying <p> without the `align` attribute resets the line to left alignment.
`mode`	No	true	`wrap | nowrap` Specifies the text wrapping mode to use. • If you specify the `mode` attribute, the device applies that mode. • If you do not specify the `mode` attribute, the device applies the last-specified `mode` value. If no previous <p> element exists, the device applies the default mode (`wrap`).

Syntax

```
<oui:p align="alignment" mode="wrapmode">
    content
</oui:p>
```

<param>

Description

The <param> tag specifies a parameter for use with a <method> tag. See <rendering_directive> for more information.

Attributes

Table 1-49. <param> tag attributes

Name	Required	rtexprvalue	Description
name	Yes	true	Specifies the variable name.
value	Yes	true	Specifies the value of the variable specified in the name attribute.

Syntax

See <rendering_directive>.

Example

See the <rendering_directive> tag for an example.

<pre>

Description

The <pre> tag is a direct counterpart of the WML <pre> element. It tells visual user agents that the enclosed text is preformatted. When handling preformatted text, user agents may:

- Leave white space intact
- Render text with a fixed-pitch font
- Disable automatic word wrap

Attributes

None

Syntax

```
<oui:pre>
   content
<oui:pre>
```

In the syntax for this tag, content can be any valid combination of the following tags:

- <oui:a>
- <oui:anchor>
- <oui:b>
- <oui:br>
- <oui:do>
- <oui:em>
- <oui:i>
- <oui:input>
- <oui:select>
- <oui:strong>
- <oui:u>

<picker>

Description

The <picker> tag presents a set of options from which the user. You can think of <picker> as a <select> construct with the sole purpose of picking an item out of a list.

Attributes

Table 1-50. <picker> tag attributes

Name	Required	rtexprvalue	Description
name	No	true	Specifies the name of the variable in which the device stores the values associated with the options the user chooses. The value associated with each option comes from the <option> element value attribute. The values in the specified variable determine the default selections when the device displays the <picker> element. If the variable has no value, the device sets it to the values specified for the default attribute. If you do not specify a default value, the device initializes the variable to an empty string (“”). In the case of multiple selections, the values are stored as a semicolon-separated list.
value	No	true	A string specifying the default values for the variable specified by the name attribute. If the name attribute already has a value when the user navigates to the <picker> element, the device ignores the value attribute. If the name attribute does not already have a value, the device sets it to the value specified by the value attribute.
iname	No	true	Identical to the name attribute except for the following: • The specified variable stores the index values associated with the options the user chooses. The index value associated with each option comes from its position in the <popup> list, starting with 1. If the user has not selected an option, the index value is 0. • The default value is specified by the ivalue attribute.

Table 1-50. <picker> tag attributes *(continued)*

Name	Required	rtexprvalue	Description
`ivalue`	No	true	Identical to the `value` attribute except that the specified string contains the default index values for the variable specified by the `iname` attribute.
`title`	No	true	Specifies the label that identifies the option. The Openwave Mobile Browser uses the title as the ACCEPT key label when the user selects the option. To ensure compatibility on a wide range of devices, the label should be a maximum of five characters.
`multiple`	No	true	`true | false` Specifies whether multiple selections are allowed.
`tabindex`	No	true	Specifies the position of the element with respect to other elements in terms of tabbing between them. The Openwave Mobile Browser does not currently support this attribute.

Syntax

```
<oui:picker name="variable"
            multiple="true|false"
            value="default"
            iname="index variable"
            ivalue="default">
    content
</oui:picker>
```

In the syntax for this tag, `content` is at least one `<option>` tag.

Example

The following example shows how to use the <picker> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="pickersample" title="Picker Sample">
        <oui:p>
            Favorite food:
            <oui:picker name="dish" multiple="true">
                <oui:option value="lchon">Lechon</oui:option>
                <oui:option value="sinig">Sinigang</oui:option>
                <oui:option value="humba">Humba</oui:option>
            </oui:picker>
        </oui:p>
    </oui:card>
</oui:wml>
```

<picker_card>

Description

The <picker_card> tag presents the user with a list of options. It allows the user to set a variable with a single click and then move on. This is similar to what both the <menu> and <picker> WML elements achieve. However, these two elements do not maximize usability across browsers, and <picker_card> does.

NOTE Because a <picker_card> is also a card, you can use this tag wherever you can use a WML card.

Attributes

Table 1-51. <picker_card> **tag attributes**

Name	Required	rtexprvalue	Description
id	No	true	A unique name for the card within the deck. The name acts as a fragment anchor for navigating to that card. For example, you can specify <go href="#cardname"/> to navigate to the <picker_card>.
var_name	No	true	Specifies the variable name where the device stores the value associated with the option the user chooses. The value associated with each option comes from the value attribute of the <menu_item> element.
href	No	true	Specifies the URL to open when the user picks from the selection.

Table 1-51. `<picker_card>` **tag attributes *(continued)***

Name	Required	rtexprvalue	Description
`link_text`	No	true	Specifies the label that appears in the primary softkey for Openwave text-based browsers only. To ensure compatibility with a wide range of devices, the label should be a maximum of five characters. Devices ignore this attribute if they do not support dynamic labeling.
`text`	No	true	Specifies the text that the device displays before the selection.
`title`	No	true	Specifies the title of the card. The Openwave text-based browser uses the title as the default bookmark name when the user bookmarks the card. Some devices use it for other purposes, such as pop-up tooltips. **NOTE:** Some browsers do not display the title as part of the card content. The phone manufacturer may display the title if there is sufficient screen real estate. However, you can force the card title to be displayed by using the `<rendering_directive>` tag.

Syntax

```
<oui:picker_card id="card_id"
                 var_name="WML_variable_name"
                 href="navigate_to_url"
                 link_text="label_primary_path"
                 text="text_on_top"
                 title="title">
    content
</oui:picker_card>
```

In the syntax for this tag, `content` is at least one occurrence of the `<oui:menu_item>` tag.

Example

The following example shows how to use the <picker_card> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:picker_card id="pickercard" var_name="dept"
        href="#phoneinput" link_text="Next"
        text="Which Dept.?">
        <oui:menu_item text="Planning"  value="plan" />
        <oui:menu_item text="Deployment"  value="deploy" />
        <oui:menu_item text="Finance"  value="cash" />
    </oui:picker_card>
</oui:wml>
```

<popup>

Description

The <popup> tag specifies a pop-up widget on graphical mobile browsers, but is rendered as a selection list on text-based browsers. It enhances usability by presenting the user with relevant options rather than requiring users to type the information manually.

Attributes

Table 1-52. <popup> tag attributes

Name	Required	rtexprvalue	Description
name	No	true	The name of the variable in which the device stores the values associated with the options the user chooses. The value associated with each option comes from the <option> element value attribute. The values in the specified variable determine the default selections when the device displays the <popup> element. If the variable has no value, the device sets it to the values specified for the default attribute. If you do not specify a default value, the device initializes the variable to an empty string (“”). In the case of multiple selections, the values are stored as a semicolon-separated list.
iname	No	true	Identical to the name attribute except for the following: • The specified variable stores the index values associated with the options the user chooses. The index value associated with each option comes from its position in the <popup> list, starting with 1. If the user has not selected an option, the index value is 0. • The default value is specified by the ivalue attribute.

Table 1-52. <popup> tag attributes *(continued)*

Name	Required	rtexprvalue	Description
value	No	true	A string specifying the default values for the variable specified by the name attribute. If the name attribute already has a value when the user navigates to the <popup> element, the device ignores the value attribute. If the name attribute does not already have a value, the device sets it to the value specified by the value attribute.
ivalue	No	true	Identical to the value attribute except for the following: • The specified string contains the default index values for the variable specified by the iname attribute.

Syntax

```
<oui:popup name="variable name"
           iname="variable name"
           value="default value"
           ivalue="default value">
    content
</oui:popup>
```

In the syntax for this tag, content requires that you set at least one <oui:option> tag value.

Example

The following example shows how to use the <popup> tag.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="popupsample" title="Alarms">
        <oui:p>
            <oui:popup name="alarm">
                <oui:option value="ocean">Ocean</oui:option>
                <oui:option value="birds">Birds</oui:option>
                <oui:option value="brook">Brook</oui:option>
            </oui:popup>
        </oui:p>
    </oui:card>
</oui:wml>
```

<postfield>

Description

The <postfield> tag is a direct counterpart of the WML <postfield> element.

Attributes

Table 1-53. <postfield> tag attributes

Name	Required	rtexprvalue	Description
name	Yes	true	Specifies the variable name of the data to be transmitted.
value	Yes	true	Specifies the data to be transmitted as part of the HTTP request.

Syntax

```
<oui:postfield name="name" value="value"/>
```

<prefetch>

Description

The <prefetch> tag preloads one or more decks to diminish latency for those phones that support this feature. It encapsulates the WML extension <link> element, so that you can take advantage of this feature for phones that support preloading, while not breaking the code on phones that do not support that feature.

NOTE The <prefetch> tag must be contained in the <oui:head> tag.

Attributes

Table 1-54. <prefetch> tag attributes

Name	Required	rtexprvalue	Description
href	Yes	true	Specifies the URL to be preloaded.

Syntax

```
<oui:wml>
    <oui:head>
        <oui:prefetch href="URL" />
    </oui:head>
    ...
</oui:wml>
```

Example

The following example shows how to use the <prefetch> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:head>
        <oui:prefetch href="nextemailbit.jsp" />
    </oui:head>
    <oui:card id="mop" title="Long Email message">
        <oui:p align="left" mode="nowrap">
            From: Domenico &lt;domenico@openwave.com&gt;<br/>
            To:Luca Passani &lt;luca@openwave.com&gt;<br/>
            Date:Wed,10 Oct 2001<br/>
        </oui:p>
        <oui:p mode="wrap">
            Subject: this is a very long message..<br/>
            ...
            [omitted]<br/>
            ...
        <oui:primary_path href="nextemailbbit.jsp"
            short_label="More"
            long_label="24% read. Read more" />
        </oui:p>
    </oui:card>
</oui:wml>
```

<prev>

Description

The <prev> tag is a direct counterpart of the WML <prev> element. It also specifies navigation to the previous URL in the history.

Attributes

None

Syntax

```
<oui:prev>content</oui:prev>
```

In the syntax for this tag, content represents the variables to set when opening the previous URL using the <oui:setvar> tag.

<primary_path>

Description

The <primary_path> tag is the main card activity that users generally take. It abstracts optimal navigation implementation across different devices. The <primary_path> tag is rendered to the primary softkey on Openwave browsers, and is rendered as the first link on Nokia browsers.

NOTE You can have only one primary path on a card.

Attributes

Table 1-55. <primary_path> tag attributes

Name	Required	rtexprvalue	Description
href	No	true	Specifies the URL to open when the user chooses the primary softkey or the first link for Nokia browsers.
long_label	No	true	Specifies the long label for the URL.
short_label	No	true	Specifies the short label for the URL.

Syntax

1) Simple case: simple URL

```
<oui:primary_path href="URL"
                  short_label="Softkey Label"
                  long_label="Hyperlink text">
```

2) Complex case: task

```
<oui:primary_path short_label="Softkey Label"
                  long_label="Hyperlink text">
    task
</oui:primary_path>
```

In the syntax for this tag, task must be one of the following tags: <oui:caller>, <oui:prev>, <oui:noop>, <oui:go>, or <oui:refresh>.

Example

The following examples show how to use the <primary_path> tag.

Example 1

This example shows how to use <primary_path> with a simple task.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="pp" title="Primary Path">
        <oui:p>
            Delete email message from Grzegorz?
            <oui:primary_path short_label="No">
                <oui:prev />
            </oui:primary_path>
            <oui:secondary_path short_label="Yes">
                <oui:go href="delete.jsp">
                    <oui:postfield name="emailid"
                    value="573ad8sd9f994da8798" />
                </oui:go>
            </oui:secondary_path>
        </oui:p>
    </oui:card>
</oui:wml>
```

Example 2

This example shows how to use <primary_path> with a complex task.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="pp" title="Primary Path 2">
        <oui:p>
            Your basket:<br/>
            Alice In Wonderland, by L.Caroll $$345.99
            <oui:primary_path short_label="buy"
                long_label="Proceed with purchase">
                <oui:go href="buy.jsp" method="get">
                    <oui:postfield name="itemid"
                        value="573ad8sd9f994da8798" />
                </oui:go>
            </oui:primary_path>
        </oui:p>
    </oui:card>
</oui:wml>
```

<radio>

Description

The <radio> tag specifies a radio button on graphical mobile browsers. On text-based browsers, radio buttons are rendered as a selection list from which the user can choose only one option.

Attributes

Table 1-56. <radio> tag attributes

Name	Required	rtexprvalue	Description
name	No	true	The name of the variable in which the device stores the values associated with the options that the user chooses. The value associated with each option comes from the <option> element value attribute. The values in the specified variable determine the default selections when the device displays the <radio> element. If the variable has no value, the device sets it to the values specified for the default attribute. If you do not specify a default value, the device initializes the variable to an empty string (“”). In the case of multiple selections, the values are stored as a semicolon-separated list.
iname	No	true	Identical to the name attribute except for the following: • The specified variable stores the index values associated with the options that the user chooses. The index value associated with each option comes from its position in the <radio> list, starting with 1. If the user has not selected an option, the index value is 0. • The default value is specified by the ivalue attribute.

Table 1-56. <radio> tag attributes *(continued)*

Name	Required	rtexprvalue	Description
value	No	true	A string specifying the default values for the variable specified by the name attribute. If the name attribute already has a value when the user navigates to the <radio> element, the device ignores the value attribute. If the name attribute does not already have a value, the device sets it to the value specified by the value attribute.
ivalue	No	true	Identical to the value attribute except for the following: • The specified string contains the default index values for the variable specified by the iname attribute

Syntax

```
<oui:radio name="variable name"
           iname="variable name"
           value="default value"
           ivalue="default value">
    content
</oui:radio>
```

In the syntax for this tag, content must contain at least one <oui:option> tag attribute.

Example

The following example shows how to use the <radio> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="radiosample" title="Alarms">
        <oui:p>
            <oui:radio name="alarm">
                <oui:option value="ocean">Ocean</oui:option>
                <oui:option value="birds">Birds</oui:option>
                <oui:option value="brook">Brook</oui:option>
            </oui:radio>
        </oui:p>
    </oui:card>
</oui:wml>
```

<raw_mode>

Description

The <raw_mode> tag specifies that the content of the tag should be rendered as written. Ideally, OUI contains everything that you need to build good wireless applications. There may be cases, however, in which you need to use parts of WML in ways that are not natively supported by the OUI WML tag library. With the <raw_mode> tag, you can specify WML that is to be rendered without OUI translation.

Attributes

Table 1-57. <raw_mode> tag attributes

Name	Required	rtexprvalue	Description
name	Yes	true	Specifies the element to be rendered without OUI translation.

Syntax

```
<oui:raw_mode element="OUI element">
    any valid combination of WML tags (no OUI tags)
</oui:raw_mode>
```

In the syntax for this tag, the OUI element can be any of the following OUI tags:

- anchor
- button
- card
- check
- do
- head
- img
- input
- menu
- onevent
- p
- picker
- popup
- radio
- table
- template
- timer

NOTE The content can be any ASCII content. Well-formed WML content is no longer guaranteed.

Example

The following examples show how to use the <raw_mode> tag:

Example1

This example shows how to render images on softkeys.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="dosample" title="Softkey Img">
        <oui:raw_mode element="do">
            <do type="accept">
                <go href="#next"/>
                    <img alt="Yes"
                        src="http://127.0.0.1/test/
                        images/portofino.wbmp"
                        localsrc="checkmark1"/>
            </do>
        </oui:raw_mode>
        <oui:raw_mode element="do">
            <do type="options">
                <go href="# cancel"/>
                    <img alt="No"
                            src="http://127.0.0.1/test/
                            images/manila.wbmp"
                            localsrc="cross1"/>
            </do>
        </oui:raw_mode>
        <oui:p>
            This shows images on soft keys
        </oui:p>
    </oui:card>
</oui:wml>
```

Example 2

This example shows how to use <raw_mode> with a conditional tag, such as <if>, to ensure that the code does not break on browsers that do not support certain tags.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="raw" title="Raw Mode">
        <oui:if special="allopenwave">
            <oui:raw_mode element="do">
            <do type="accept" label="Spawn">
            <spawn href="http://www.openwave.com" />
            </do>
        </oui:raw_mode>
    </oui:if>
        <oui:p align="left">
            Openwave browsers with an Openwave gateway
            will have a do element which implements the 'spawn'
            extension....<br/>
        </oui:p>
    </oui:card>
</oui:wml>
```

<refresh>

Description

The <refresh> tag is a direct counterpart of the WML <refresh> element.

Attributes

None

Syntax

```
<oui:refresh>content</oui:refresh>
```

In the syntax for this tag, content represents the variables to refresh as specified by the <oui:setvar> tag.

<rendering_directive>

Description

A directive is similar to a compiler option. Directives tell OUI to change from its default behavior in some specific cases when rendering static WML for a browser.

Attributes

Table 1-58. <rendering_directive> tag attributes

Name	Required	rtexprvalue	Description
apply_to	Yes	true	`card | body_pager | deck | device_context | template` Specifies the context to which the directive should be applied.

Syntax

1) Simple case: No parameters

```
<oui:rendering_directive apply_to="OUI Object Name">
    <oui:method name="directive name" />
</oui:rendering_directive>
```

Table 1-59. <rendering_directive> tag objects and directives

OUI Object Name	Directive Name
body_pager	• enforce_title: Puts the title in the body of the card if the device does not support the title attribute of the <card> element. • enforce_navigation_with_links: Enforces all navigation to be rendered using <anchor> tags, regardless of the device.
card	• enforce_title: Puts the title in the body of the card if the device does not support the title attribute of the <card> element. • enforce_navigation_with_links: Enforces all navigation to be rendered using <anchor> tags, regardless of the device • enforce_logical_back: Redefines the <prev> task by redirecting the user to another deck or card when the user navigates to the current card in a backward direction. Useful for devices that do not properly redefine the <prev> task. This method requires the backurl parameter, which contains the destination URL. See "<param>" on page 79 for information on how to specify a parameter.
deck	• disable_nokia_back_navigation: Tells OUI not to automatically define a <prev> task for Nokia browsers.
device_context	• disable_extensions: Instructs OUI to use the WML DTD instead of the Openwave DTD for the WML deck thereby disabling all Openwave extensions. All tags in the deck are rendered using only WML elements. • enable_extensions: Instructs OUI to use the Openwave DTD instead of the WML DTD, thereby enabling all Openwave extensions to WML. • optimize_for_speed: Instructs OUI to skip some automatic validations checks OUI performs internally. If you are confident that your OUI object model is correct, you can use this directive to achieve a slight performance improvement. • disable_content_type: OUI normally produces a MIME content type before it delivers the actual WML to the gateway and device. In all of those cases when you want to take control of the HTTP header, you can do it as you typically would with a servelet, but you may want to avoid having OUI generate the MIME type.
template	• disable_nokia_back_navigation: Tells OUI not to automatically define a <prev> task inside the <template> tag for Nokia browsers. If you have defined a <template> tag, all the contents of the <template> tag will be rendered without the additional <prev> task that is usually added by OUI during translation.

2) Complex case: Parameters to rendering directive required

```
<oui:rendering_directive apply_to="WAOM Object Name">
    <oui:method name="directive name" >
        <oui:param name="parameter name"
            value="parameter value" />
    </oui:method>
</oui:rendering_directive>
```

Table 1-60. <rendering_directive> tag objects and directives

OUI Object Name	Directive Name	Parameter(s)	Parameter Value
card (<form>, <picker_card>)	enforce_logical_back	backURL	URL to be opened when the user navigates to the card in a backward direction.

Example

The following examples show how to use the <rendering_directive> tag:

Example1

This example shows how to use the enforce_title directive.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="pharmacy" title="Churchill Pharmacy">
        <oui:rendering_directive apply_to="card">
            <oui:method name="enforce_title" />
        </oui:rendering_directive>
        <oui:p>
            8 Ladybrook Square
            Ladybrook Square
            Hemsel
            Nottinghamshire
            N52 7UL
            <br/>
            <a href="map.jsp?id=32647287328"
                title="map">See map</a>
        </p>
        <p mode="nowrap">
            Tel:(01452)9344615582<br/>
            Fax:(01452)2355810158<br/>
            info@openwave.com
            <oui:primary_path href="top.jsp" short_label="Main"
                long_label="Back to main page" />
        </oui:p>
    </oui:card>
</oui:wml>
```

Example 2

This example shows how to use the `enforce_logical_back` directive.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="card1" title="Step 1">
        <oui:p align="left">
            Card 1
            <oui:primary_path href="#card2"
                short_label="Card2" />
        </oui:p>
    </oui:card>
    <oui:card id="card2" title="Step 2">
        <oui:rendering_directive apply_to="card">
            <oui:method name="enforce_logical_back" >
                <oui:param name="backURL" value="#card1" />
            </oui:method>
        </oui:rendering_directive>
        <oui:p align="left">
            Card 2
            <oui:primary_path href="#card3"
                short_label="Card3" />
        </oui:p>
    </oui:card>
    <oui:card id="card3" title="Step 3">
        <oui:p align="left">
            Card 3
            <oui:primary_path href="http://www.openwave.com/"
                short_label="OPWV" />
        </oui:p>
    </oui:card>
</oui:wml>
```

<secondary_path>

Description

The <secondary_path> tag encapsulates an important but not primary path activity for a card. These are activities that many users, but not the majority, often perform. This tag abstracts how navigation can be optimally implemented across different devices. The <secondary_path> tag is rendered to the secondary softkey on Openwave browsers and as a link on Nokia browsers.

Attributes

Table 1-61. <secondary_path> tag attributes

Name	Required	rtexprvalue	Description
href	No	true	Specifies the URL to open when the user chooses the secondary softkey for Openwave browsers or hyperlink for Nokia browsers.
long_label	No	true	Specifies the long label for the link.
short_label	No	true	Specifies the short label for the link.

Syntax

Simple case: simple URL

```
<oui:secondary_path href="URL"
                    short_label="Softkey Label"
                    long_label="Hyperlink text">
```

Complex case: task

```
<oui:secondary_path short_label="Softkey Label"
                    long_label="Hyperlink text">
    task
</oui:secondary_path>
```

In the syntax for this tag, task is one of the following tags: <oui:caller>, <oui:prev>, <oui:noop>, <oui:go>, or <oui:refresh>.

Example

The following examples show how to use the <secondary_path> tag.

Example 1

This example shows a secondary path with a <go> task and a <postfill>.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="sp" title="Secondary path">
        <oui:p>
            Delete email message from Grzegorz?
            <oui:primary_path short_label="No">
                <oui:prev />
            </oui:primary_path>
            <oui:secondary_path short_label="Yes">
                <oui:go href="delete.jsp">
                    <oui:postfield name="emailid"
                        value="573ad8sd9f994da8798" />
                </oui:go>
            </oui:secondary_path>
        </oui:p>
    </oui:card>
</oui:wml>
```

Example 2

This example shows multiple secondary paths with a <prev> task.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="sp" title="Secondary Path">
        <oui:p>
            Multiple Secondary paths<br/>
            Try again or move to next step?
            <oui:primary_path href="next.jsp"
                short_label="Next" />
            <oui:secondary_path href="tryagain.jsp"
                short_label="Again" />
            <oui:secondary_path short_label="Back">
                <oui:prev />
            </oui:secondary_path>
        </oui:p>
    </oui:card>
</oui:wml>
```

<setvar>

Description

The <setvar> tag is a direct counterpart of the WML <setvar> element.

Attributes

Table 1-62. <setvar> tag attributes

Name	Required	rtexprvalue	Description
name	Yes	true	Specifies the variable name.
value	Yes	true	Specifies the value to be assigned to the variable.

Syntax

```
<oui:setvar name="variable name" value="value"/>
```

<side_path>

Description

The <side_path> tag specifies an activity that most users do occasionally. It abstracts how navigation can be optimally implemented across different devices. The <side_path> tag is rendered to the secondary soft key on Openwave browsers and as a link in the Options menu on Nokia browsers.

Attributes

Table 1-63. <side_path> tag attributes

Name	Required	rtexprvalue	Description
href	No	true	Specifies the URL to open when the user chooses the secondary softkey for Openwave browsers or the link from the Options menu for Nokia browsers.
long_label	No	true	Specifies the long label for the link.
short_label	No	true	Specifies the short label for the link.

Syntax

Simple case: simple URL

```
<oui:side_path href="URL"
               short_label="Softkey Label"
               long_label="Hyperlink text">
```

Complex case: task

```
<oui:side_path short_label="Softkey Label"
               long_label="Hyperlink text">
    task
</oui:side_path>
```

In the syntax for this tag, task is one or more of the following tags: <oui:caller>, <oui:prev>, <oui:noop>, <oui:go>, or <oui:refresh>.

Example

The following example shows how to use the <side_path> tag.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="pp" title="Side Path">
        <oui:p>
            Side paths are for activities users hardly ever
            do when using a mobile application.
            <oui:primary_path href="pp.jsp"
                short_label="PPath" />
            <oui:secondary_path href="sp.jsp"
                short_label="SPath" />
            <oui:side_path href="sidep.jsp"
                short_label="SidePath" />
        </oui:p>
    </oui:card>
</oui:wml>
```

<small>

Description

The <small> tag is a direct counterpart of the WML <small> element. It specifies small text type.

Attributes

None

Syntax

```
<oui:small>text</oui:small>
```

In the syntax for this tag, text is the text to display in small type.

Description

The tag is a direct counterpart of the WML element. It specifies strongly emphasized text type.

Attributes

None

Syntax

```
<oui:strong>text</oui:strong>
```

In the syntax for this tag, `text` is the text to display in strongly emphasized type.

<table>

Description

The <table> tag encapsulates the WML <table> element. It is useful for building tables without breaking the rendering on phones that do not support tables. There are three ways of interpreting the meaning of information laid out in a table: Row logic, column logic, and matrix logic. OUI fully supports tables of the first two types and attempts to display tables with matrix logic.

Attributes

Table 1-64. <table> **tag attributes**

Name	Required	rtexprvalue	Description
logic	No	true	row | column | matrix Specifies the table logic as follows: row: With row logic, information displayed in a table makes sense when interpreted on a row basis. column: With column logic, information displayed in a table makes sense when interpreted on a columnar basis. matrix: With matrix logic, information displayed in a table makes sense when interpreted on both dimensions.
columns	Yes	true	Specifies the number of columns for the row set. This value cannot be zero.
title	No	true	Specifies a label for the table.
align	No	true	left | right | center Specifies text alignment relative to the column. If you do not specify the align attribute, the text is automatically left aligned.

Syntax

```
<oui:table logic="logic" columns="number of columns">
    <oui:tr>
        <oui:td>cell 1.1 </oui:td>
        <oui:td>cell 1.2</oui:td>
    </oui:tr>
        ...
    <oui:tr>
        <oui:td>cell 2.1</oui:td>
        <oui:td>cell 2.2</oui:td>
    </oui:tr>
</oui:table>
```

In the syntax for this tag, the value of the `columns` attribute must match the effective number of columns. The `logic` attribute can be `row`, `column`, or `matrix`.

Example

The following examples show how to use the `<table>` tag.

Example 1

This example produces a table with row logic.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="starttable" title="FIFA World Cup">
        <oui:p align="center">
            FIFA World Cup European Qualifying<br/>
            Group Eight:
        </oui:p>
        oui:p align="left">
            <oui:table logic="row" columns="2">
                <oui:tr>
                    <oui:td>Team </oui:td>
                    <oui:td>Pts</oui:td>
                </oui:tr>
                <oui:tr>
                    <oui:td>1.Italy </oui:td>
                    <oui:td>20</oui:td>
                </oui:tr>
                <oui:tr>
                    <oui:td>2.Romania</oui:td>
                    <oui:td>16</oui:td>
                </oui:tr>
                <oui:tr>
                    <oui:td>3.Georgia</oui:td>
                    <oui:td>10</oui:td>
                </oui:tr>
                <oui:tr>
                    <oui:td>4.Hungary</oui:td>
                    <oui:td>8</oui:td>
                </oui:tr>
                <oui:tr>
                    <oui:td>5.Lithuania</oui:td>
                    <oui:td>2</oui:td>
                </oui:tr>
            </oui:table>
            <oui:primary_path href="viewgroup.jsp?id=9"
                short_label="Next" long_label="See Group 9" />
            <oui:secondary_path href="eurozone.jsp"
                short_label="Up" long_label="Europeanzone" />
            <oui:secondary_path href="viewgroup.jsp?id=7"
                short_label="Back" long_label="See Group 7" />
        </oui:p>
    </oui:card>
</oui:wml>
```

Example 2

This example produces tables with column and matrix logic.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="eattb2" title="Tables">
        <oui:p>
            Table with column logic
            <oui:table logic="column" columns="2">
                <oui:tc>
                    <oui:td>Pasta</oui:td>
                    <oui:td>100 DKK</oui:td>
                </oui:tc>
                <oui:tc>
                    <oui:td>Lobster</oui:td>
                    <oui:td>236 DKK</oui:td>
                </oui:tc>
            </oui:table>
        </oui:p>
        <oui:p>
            Table with matrix logic
            <oui:table logic="matrix" columns="3">
                <oui:th>
                    <oui:td></oui:td>
                    <oui:td>High</oui:td>
                    <oui:td>Low</oui:td>
                </oui:th>
                <oui:tr>
                    <oui:td>Milan</oui:td>
                    <oui:td>34</oui:td>
                    <oui:td>21</oui:td>
                </oui:tr>
                <oui:tr>
                    <oui:td>Rome</oui:td>
                    <oui:td>32</oui:td>
                    <oui:td>25</oui:td>
                </oui:tr>
            </oui:table>
        </oui:p>
    </oui:card>
</oui:wml>
```

<task_menu>

Description

The <task_menu> tag specifies a set of menu options associated with the secondary softkey or the OPTIONS key. When the user selects the secondary softkey or OPTIONS key on graphical browsers, a pop-up menu appears above the key. On the Nokia and text-based browsers, the menu is shown in another card. This mechanism replaces the traditional way of supporting multiple paths on text-based browsers.

On Openwave browsers, the task menu options can be selected either by pressing the key accelerator associated with the item in the pop-up menu, or by using the up/down navigation to highlight one of the options and then pressing the ACCEPT key.

Attributes

Table 1-65. <task_menu> tag attributes

Name	Required	rtexprvalue	Description
long_label	No	true	Specifies the long label for the link.
short_label	No	true	Specifies the short label for the link.

Syntax

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="tm" title="Task Menu">
        <oui:p>
            ... card content ...
        </oui:p>
        <oui:task_menu short_label="Softkey Label"
            long_label="Hyper Link Text">
            <oui:menu_item href="URL" text="Menu Item Text"/>
            <oui:menu_item href="URL" text="Menu Item Text"/>
                ...
            <oui:menu_item href="URL" text="Menu Item Text"/>
            </oui:task_menu>
    </oui:card>
</oui:wml>
```

Example

The following example shows how to use the <task_menu> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="Inbox" title="Openwave Email">
        <oui:p mode="nowrap">
            <oui:menu>
                <oui:menu_item href="ema.jsp?344b53"
                    title="View"
                    text="Luca Passani:Portofino" />
                <oui:menu_item href="ema.jsp?345a29"
                    title="View" text="Lars Knudesn:Fixed!" />
                <oui:menu_item href="ema.jsp?345a49"
                    title="View"
                    text="Grzegorz Cipiel:RE:Installer...." />
                <oui:menu_item href="ema.jsp?345a45"
                    title="View"
                    text="Jaycelle Lao:Urgent Issue  " />
                <oui:menu_item href="ema.jsp?345a59"
                    title="View"
                    text="Srini Bhag..:Vegetarian recipies." />
                <oui:menu_item href="ema.jsp?345a59"
                    title="View"
                    text="Raj: Re:I need an answer" />
                <oui:menu_item href="ema.jsp?345a23"
                    title="View"
                    text="Heidi Gilstad: RE:Copenhagen" />
                <oui:menu_item href="ema.jsp?345a24"
                    title="View"
                    text="Sandra Bella:is Tivoli open?" />
                <oui:menu_item href="ema.jsp?345a25"
                    title="View"
                    text="Andrea Masella:Stasera? " />
            </oui:menu>
        </oui:p>
        <oui:task_menu short_label="Menu"
            long_label="Show Menu">
            <oui:menu_item href="#compose" text="Compose New"/>
            <oui:menu_item href="#viewfldr"
                text="View Folders"/>
            <oui:menu_item href="#newfldr"
                text="Create New Folder"/>
            <oui:menu_item href="#signout" text="Sign Out"/>
        </oui:task_menu>
    </oui:card>
</oui:wml>
```

<tc>

Description

The <tc> tag specifies a column of data in a table. It is a container that holds a single table column. Table columns can be empty. The number of <tc> tags should match the number of columns in the table. See the columns attribute of the <table> tag.

Attributes

None

Syntax

```
<oui:tc>
    content
</oui:tc>
```

In the syntax for this tag, content specifies the data for the column as denoted by one or more <oui:td> tags.

NOTE You can have an empty <td> tag, but you must specify the same number of <td> tags in every <tc> (column) tag.

Example

See the <table> tag for an example.

<td>

Description

The <td> tag is a direct counterpart of the WML <td> element. It is a container that holds a single table cell data within a table row or column. Table data cells can be empty. The user agent should attempt to deal with multiple-line data cells that may result from using images or line breaks.

Attributes

None

Syntax

```
<oui:td>content</oui:td>
```

In the syntax for this tag, content represents text inside the table cell.

NOTE The <oui:img> and <oui:anchor> elements in a table cell will be supported in a future release of OUI.

Example

See the <table> tag for an example.

<template>

Description

The <template> tag is a direct counterpart of the WML <template> element. Like the WML <template> element, it defines deck-level event bindings, that is, characteristics that apply to all cards in the deck. You can override these characteristics for a particular card by specifying the same event bindings in the <card> definition.

Attributes

Table 1-66. <template> tag attributes

Name	Required	rtexprvalue	Description
onenterforward	No	true	Specifies the URL to open if the user navigates to a card through a <go> task. This attribute is an abbreviated form of the <onevent> element.
onenterbackward	No	true	Specifies the URL to open if the user navigates to a card through a <prev> task. This attribute is an abbreviated form of the <onevent> element.
ontimer	No	true	Specifies the URL to open if a specified <timer> element expires. This attribute is an abbreviated form of the <onevent> element.

Syntax

```
<oui:wml>
    <oui:template onenterforward="url"
                  onenterbackward="url"
                  ontimer="url">
        content
    </oui:template>
    card content
</oui:wml>
```

In the syntax for this tag, content inside the <template> tag represents the general action to take when <oui:do> or <oui:onevent> events occur.

<th>

Description

The <th> tag specifies a table header that gives column names to a table. This tag can only be used in tables that use matrix logic.

Attributes

None

Syntax

```
<oui:th>
    content
</oui:th>
```

In the syntax for this tag, content specifies the data for the column using the <td> tag.

NOTE You can have an empty <td> tag, but you must specify one <td> tag for each column in the table.

Example

See the <table> tag for an example.

<then>

Description

The <then> tag specifies the code that should be rendered if the conditions defined as part of the <oui:condition> or <oui:conditionlist> tag are satisfied.

Syntax

```
<oui:if>
    conditions (<oui:condition> or <oui:condition_list>)
    <oui:then>
        any valid combination of OUI tags
    <oui:then>
    <oui:else>
        any valid combination of OUI tags
    </oui:else>
</oui:if>
```

Example

See <condition_list> for an example.

<timer>

Description

The <timer> tag is a direct counterpart of the WML <timer> element.

Attributes

Table 1-67. <timer> tag attributes

Name	Required	rtexprvalue	Description
name	No	true	The name of the variable in which the device stores the timer value. If the variable has no value when the timer is initialized, the device sets it to the value specified by the value attribute. The device sets this variable to either the current timer value when the user exits the card or to 0 if the timer expires.
value	No	true	A string specifying the value for the variable specified by the name attribute. You must specify <timer> values in units of 1/10 seconds. For example, a value of 100 equals 10 seconds. Specifying a value of 0 disables the timer. If the name attribute already has a value when the timer is initialized, the device ignores the default attribute. If the name attribute does not already have a value, the device sets it to the value specified by the value attribute.

Syntax

```
<oui:timer name="timer variable" value="value"/>
```

<tr>

Description

The <tr> tag is a direct counterpart of the WML <tr> element. It is a container that holds a single table row. Table rows can be empty.

Attributes

None

Syntax

```
<oui:tr>
    content
</oui:tr>
```

In the syntax for this tag, content represents table cells defined using the <td> tag

NOTE You can have an empty <td> tag, but you must specify the same number of <td> tags for every row.

Example

See the <table> tag for an example.

<u>

Description

The <u> tag is a direct counterpart of the WML <u> element. It specifies underlined text.

Attributes

None

Syntax

```
<oui:u>text</oui:u>
```

In the syntax for this tag, text is the text to display in underlined type.

<unset>

Description

The <unset> tag clears the value of a variable.

Attributes

Table 1-68. <unset> **tag attributes**

Name	Required	rtexprvalue	Description
name	Yes	true	Specifies the name of the variable to be cleared.

Syntax

```
<oui:unset name="WML variable" name/>
```

Example

The following examples show how to use the <unset> tag:

Example1

This example shows how to use the <unset> tag to clear the contents of a WML variable.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:card id="setsample" title="Setvar">
        <oui:p>
            This card sets the variable
        </oui:p>
        <oui:primary_path short_label="Unset">
            <oui:go href="#unsetsample">
                <oui:setvar name="fname" value="rowena"/>
            </oui:go>
        </oui:primary_path>
        <oui:secondary_path href="#exit" short_label="Cancel"/>
    </oui:card>
    <oui:card id="unsetsample" title="Unset">
        <oui:unset name="fname"/>
        <oui:p>
            Hi $fname
            This card unsets the variable
        </oui:p>
        <oui:primary_path short_label="OK">
            <oui:go href="#card3"/>
        </oui:primary_path>
        <oui:secondary_path href="#exit" short_label="Cancel"/>
    </oui:card>
    <oui:card id="card3" title="Test Unset">
        <oui:p>
            Hi $fname!
            You're in card3.
        </oui:p>
        <oui:primary_path href="#nextcard" short_label="OK"/>
        <oui:secondary_path href="#exit" short_label="Cancel"/>
    </oui:card>
</oui:wml>
```

Example 2

This example shows how to use the <unset> tag to initialize the WML variables when the user enters a card or form.

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml>
    <oui:form id="train" title="Find your train">
        <oui:unset name="startstation" />
        <oui:unset name="endstation" />
        <oui:unset name="when" />
        Hallo everyone!
        Start in (min 3):
        <oui:input type="text" size="6" title="from"
            name="startstation" value="" />
        End in (min 3):
        <oui:input type="text" size="6" title="to"
            name="endstation" value="" />
        When:
        <oui:picker name="when" title="when">
            <oui:option value="3">within 3 hours</oui:option>
            <oui:option value="9">within 9 hours</oui:option>
            <oui:option value="0">whenever</oui:option>
        </oui:picker>
        <oui:primary_path short_label="Find"
            long_label="Find station">
            <oui:go href="search.jsp">
                <oui:postfield name="startstation"
                    value="$startstation" />
                <oui:postfield name="endstation"
                    value="$endstation" />
                <oui:postfield name="when" value="$when" />
            </oui:go>
        </oui:primary_path>
    </oui:form>
</oui:wml>
```

<wml>

Description

The <wml> tag encapsulates the WML <wml> element, which specifies a WML deck. You can use the <oui:wml> tag to specify the time to live, the character set to use, whether the deck can be bookmarked, and the bookmark URL to use if the deck cannot be bookmarked—all without having to use cryptic code with the <meta> tag.

Attributes

Table 1-69. <wml> tag attributes

Name	Required	rtexprvalue	Description
bookmarkable	No	true	true | false Specifies whether the current deck can be bookmarked.
bookmarkURL	No	true	Specifies the bookmark URL if the current deck cannot be bookmarked.
ttl	No	true	Specifies the time to live or the length of time (in 1/10 of a second) that a device keeps the deck in cache/memory.
char_set	No	true	Specifies the character set to be used in the deck

Syntax

```
<oui:wml bookmarkable="bookmarkable"
         bookmarkURL="url"
         ttl="time to live"
         char_set="character set">
    <oui:card>
        card content
    </oui:card>
</oui:wml>
```

Example

The following example shows how to use the <wml> tag:

```
<%@ taglib uri="/WEB-INF/tld/oui.tld" prefix="oui" %>
<oui:wml ttl="30" bookmarkable="false"
    bookmarkURL="http://www.openwave.com" >
    <oui:card id="testcard" title="MyCard" >
        <oui:p>
            This is my card. If you bookmark this card,
            you'll be bookmarking www.openwave.com.
        </oui:p>
    </oui:card>
</oui:wml>
```