Generic Content Authoring Guide for WML 1.1

Version 8 February-2001

Wireless Application Protocol
Best Practices and Recommendations for authoring WML content in a generic fashion

Notice:

Disclaimer:

This is document is not a WAP Forum specification. This document is not endorsed by the WAP Forum and is informative only. This document is subject to revision or removal without notice. No part of this document may be used to claim conformance or interoperability with the WAP Forum specifications.

Contents

1. Scope

Wireless Application Protocol (WAP) is a result of continuous work to define an industry-wide specification for developing applications that operate over wireless communication networks. The scope for the WAP Forum is to define a set of specifications to be used by service applications. The wireless market is growing very quickly and reaching new customers and services. To enable operators and manufacturers to meet the challenges in advanced services, differentiation, and fast/flexible service creation, WAP defines a set of protocols in transport, session, and application layers. For additional information on the WAP architecture, refer to "Wireless Application Protocol Architecture Specification" [WAP].

This document provides developers with a content authoring guide that will enable them to develop complex WML 1.1 applications that consistently display and are usable across a wide range of WAP devices. A significant advantage to be gained from the use of this guide is the ability to create applications that do not require extensive testing effort across a large set of WAP browsers.

This guide provides a series of recommendations that aid the developer to produce effective static and dynamic content, despite the differences in available browser implementations. Recommendations are split into Category 1, which should always be followed when producing best practice generic content, and Category 2, which may not be appropriate in all situations.

This guide does not attempt to define the ‘style’ by which an application should be designed, but includes recommendations that assist the application to be useable over a wide range of devices. Full application development should of course include product usability testing and possibly corporate or other style guides to enable the complete design of an effective, customer centred application.

Issues relating to WMLScript, origin server configuration, usability methodology, and XML transformations are outside the scope of this document.

The recommendations described in this guide are **optional** and **do not preclude developers from also creating applications that maximize behaviours of individual browsers** – potentially leading to improvements in performance, usability and competitiveness. However following the recommendations will result in an application that will run, have basic usability and display satisfactorily on a wide range of browsers and can hence be considered best practice for generic applications design.

2. Document Status

This document is available online in the following formats:

PDF format at http://www.wapforum.org/.

2.1 Copyright Notice

Terms and conditions of use are available from the Wireless Application Protocol Forum Ltd. web site at http://www.wapforum.org/docs/copyright.htm.

2.2 Errata

Known problems associated with this document are published at http://www.wapforum.org/.

2.3 Comments

Comments regarding this document can be submitted to the WAP Forum in the manner published at http://www.wapforum.org/.

3. References

3.1 Informative References

[RFC2396] "Uniform Resource Identifiers (URI): Generic Syntax", T. Berners-Lee, et al., August 1998.
URL: http://www.ietf.org/rfc/rfc2396.txt

[WAP] "Wireless Application Protocol Architecture Specification", WAP Forum, 30-Apr-1998.
URL: http://www.wapforum.org

[WML] "Wireless Markup Language Specification", WAP Forum, 04-Nov-1999.
URL: http://www.wapforum.org

4. Definitions and Abbreviations

4.1 Definitions

The following are terms used throughout this document.

Browser – a browser is any software or device that interprets resources for presentation to users. This may include textual browsers, voice browsers, etc.

Card – a single WML unit of navigation and user interface. May contain information to present to the user, instructions for gathering user input, etc.

Deck – a collection of WML cards. A WML deck is also an XML document.

Device – a network entity that is capable of sending and receiving packets of information and has a unique device address. A device can act as both a client and a server within a given context, or across multiple contexts. For example, a device can service a number of clients (as a server) while being a client to another server.

PDA – a small, portable, personal device that is designed to run a variety of applications, and accept extended textual input. It is generally not capable of placing or receiving phone calls.

Test Example – one or more WML decks that are intended to demonstrate the breadth of ways to interpret a particular aspect of WML.

WML – the Wireless Markup Language is a hypertext markup language used to represent information for delivery to a narrow-band device, eg. a mobile phone.

WMLScript – a scripting language used to program the mobile device. WMLScript is an extended subset of the JavaScript™ scripting language.

XML – the Extensible Markup Language is a World Wide Web Consortium (W3C) recommended standard for Internet markup languages, of which WML is such a language. XML is a restricted subset of SGML.

4.2 Abbreviations

For the purposes of this guide, the following abbreviations apply.

GIF	Graphics Interchange Format
PDA	Personal Digital Assistant
SDK	Software Development Kit
URL	Uniform Resource Locator [RFC1738]
WAP	Wireless Application Protocol [WAP]
WBMP	Wireless Bitmap
WTA	Wireless Telephony Applications
XML	Extensible Markup Language

5. Background

5.1 WAP Application Development

WAP content is developed using the WML language. This document specifically refers to the WML version 1.1 specification.

Writing WML code for an application does not guarantee that the application will either appear or behave in the same way on all browsers. Each browser is a contract with the WML specification and even when the browser complies with the specification it can still differ from another browser in fulfilling that contract. An objective measure does not exist which says which manner of interpreting WML is superior to another. Nevertheless, the differences that arise are noticeable and are sometimes contrary to customer expectations (keeping in mind that customers will often bring to the task previous experiences working at similar interfaces). These differences in browser rendering may adversely influence customer perceptions and behavior.

WML version 1.1	**Response**
Non-Compliance	Device manufacturer to comply
Ambiguity	Tighten WML specification
Different rendering	Style guidelines

Table 1: Sources of variation in Browser display and behavior

Table 1 shows three ways in which the same WML code might lead to differences in display and behavior on different browsers. Non-compliance refers to the case where the browser of a WAP device renders WML code in a way that is not stipulated by the WML specification. The device manufacturer should address this. Ambiguity refers to instances where the WML specification is open to many different but valid interpretations by the device manufacturers. This area may, in the future, be improved by tightening the WML specification. Finally, when the same WML code is rendered across different browsers, each user may perceive the application quite differently. These variations may be viewed as legitimate and harmless so long as they do not interfere with usability. Vendors may use these differences to differentiate themselves from the rest of the market.

The Generic Content Authoring Guide addresses these sources of variation through a set of recommendations. These recommendations are about writing WML code to minimize the variant behaviour that harms application usability across a range of browsers.

5.2 Using test examples to arrive at recommendations

A number of test examples were prepared to assist in formulating these guidelines. Test examples were formed by using existing literature and by using WML 1.1 from a developer perspective. A ‘test example’ consisted of a few lines of WML 1.1 code aimed to assess a ‘tag’ and the behavior of such tags when associated in a deck of cards, from a developers’ perspective. The test examples were run on a number of different phones and phone simulators. Test examples were also applied to PDAs. The observations resulting from running the test examples on a number of browsers were then distilled into this guide. Recommendations were formulated that were generic and device independent, but also concrete and specific.

The goal for this guide was to establish generic usability/design recommendations across different clients, and not to define how vendors should make their browsers behave.

Only devices that accepted WML v1.1 were considered for testing. WML Script and other WTA features were not considered.

The test examples are available separately and it is recommended to run at least some of the test examples on a variety of browsers/simulators to understand how the recommendations were made.

Simulators	Nokia . WAP Tool Kit V1.3b Ericsson . WAP IDE 2.0B8 Openwave . UP.SDK v3.2
Phones	Ericsson R380S, R320S, Nokia 7110
PDA	Ericsson MC218 v1.13

Table 2: Devices used for testing

Simulators	Openwave . UP.SDK v4.0
Phones	Siemens S35 Phillips Azalis/Ozeo/Xenium Mitsubishi Trium
PDA	Palm IIIx with AU-System browser

Table 3: Devices that impacted the recommendations, but were not extensively tested

6. Design Orientation

6.1 Design Attributes

An established way of orienting design activity is one based around the customer. Such a process encourages the developer to view their product from the perspective of the customer. The product should be simple to use, easy to learn, pleasant to view and satisfying. We define the following attributes that should be considered in the design of WAP applications:

- 'Usability' is defined as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" (P2; ISO 9 241-11:1998(E)).
- Effectiveness means accuracy and completeness with which customers achieve specified goals (e.g. obtaining a specific share price from a stock exchange).
- Efficiency means resources expended in relation to the accuracy and completeness with which customers achieve goals (e.g. making fewer key presses to arrive at a destination; taking less time to achieve a goal, costing less to obtain some specific information).
- Satisfaction means freedom from discomfort, and positive attitudes towards the use of the product. Pleasure from using a product is often a powerful motivator for its continued use.
- Context of use refers to customers, their tasks, equipment and the physical and social environment in which their activity takes place. When designing applications choose a typical customer and model how they use the product.

6.2 Design Hierarchy

In every design activity there are a number of objectives to satisfy, and here the focus is in reducing the variance in the ways an application is presented to users. Table 4 shows that the burden of work should be shifted from the customer up the hierarchy towards the WAP server, where it is better to handle it. However, a great deal of effort is required to address variance at the highest levels as many parties have already deployed equipment, but the developer is often a single party and able to change application behaviour. The developer's role, through the use of a specification language, is to:

- provide customers with an application whose behaviour is consistent at the interface; and
- meet customers' expectations of using such an application.

A generic WAP content authoring guide for application developers has advantages for all stakeholders as it addresses the high degree of variance in interfaces.

Entities in application delivery	**Gain from addressing variance**	**Effort required to address variance**
WAP server	Promotes value	Most effort
WAP Handset	Promotes accessibility	
Developer	Promotes consistency and reduces development costs	
Customer / User	Promotes usability and satisfaction	Least effort

Table 4: WAP Design Hierarchy

6.3 Device Constraints

A WAP device generally takes the form of a phone with a small browser. However, other products with WAP capabilities, e.g. Personal Digital Assistants (PDAs), may become increasingly common in the market place. Thus a guide to development should consider the following limitations:

- The device is primarily a phone (small and lightweight, and therefore portable) but can also be in other forms.
- A small (black and white) display is used primarily for retrieving information.
- The product may be used in physical and social environments, which may influence interaction.
- Entering text can be cumbersome and is typically slower than using a desktop keyboard.

6.4 Design Principles

Developing for WAP is an opportunity to apply and extend lessons learnt from previous decades of interface design development. A number of principles have been distilled for the present purpose:

- Know the customer population – design for real people.
- Focus on the context and instances of use as a guide to understanding how customers will interact with the application and device.
- Reduce cognitive load – unnecessary thinking, remembering, having to make inferences, having to deduce, cognitive dependencies. Don’t expect customers to know about ‘xyz’ to achieve ‘abc’.
- Engineer for errors – human error should not be an excuse for failing to use the product as designed. The application should accommodate all user actions.
- Consider the case where particular browser features cannot be fulfilled.
- Maintain consistency and clarity.
- Facilitate navigation (“how do I get to …?”, “How do I get back to the start?”).

7. Best Practice and Recommendations

7.1 Objective

The objective of this document is to provide generic content authoring and usability/design guidelines, that are comprehensive (but make no claim to being exhaustive) and yet device independent. The guide does not require nor define how vendors should make their browsers behave.

Recommendations fall into two categories. Category 1 recommendations should be followed if the developer desires for the application to work across a range of browsers. Category two recommendations should be followed most of the time, but there may be cases where it is prudent to disregard the recommendation.

In order to facilitate the use and application of the guidelines the recommendations have been grouped in two ways. Within each group, recommendations are sorted with Category 1 recommendations presented first:

1. Each tag by **functional grouping** – Navigation, Form, Appearance and Procedure.
2. By **tag** – `<anchor>`, `<card>`, `<do>`, `<fieldset>`, `<image>`, `<input>`, `<miscellaneous>`, `<onevent>`, `<optgroup>`, `<paragraph>`, and `<select>`.

Justifications for each recommendation can be found in **Annex 1**. Test examples which were used to arrive at the recommendations for each tag are also referenced.

Internationalisation Note: These recommendations have been written with Western alphabets and the English language in mind. However, the recommendations ought to be interpreted with the following considerations in mind:

- Non-western characters will usually fill the equivalent space of two western characters, and should count as such when interpreting recommendations involving amounts of characters. For example, a row in a table should not be more than 6 Asian characters long.
- All words used for navigation and information ideally should be in the same language. For example, in a French application the word “Retour” should be used as the label for an event if it has `<prev>` as its action.

7.2 Design Recommendations by Function

7.2.1 Navigation

When the purpose of the card is to retrieve another card.

General recommendations:

Category 1 Recommendations	• Different decks should have different URLs.
Category 2 Recommendations	• A collection of items should be arranged in a hierarchy so that its dimension is greater in breadth than depth.
Test examples	anchor4.cgi

Recommendations relevant to the tag **`<anchor>`**:

Category 1 Recommendations	• Where anchored text appears, there should not be any other anchored text or images on that line, i.e. anchored elements are separated from other anchored elements by `<br/>` or `<p>` tags. • Anchored text should be kept as short as possible, and should not be longer than one or two words unless required for understanding. • The continuation of a list of items (anchored text, select items, etc.) should be indicated to the user – either through the use of language (e.g. “8 Items”), or with syntax through the list. • A list of items used for input or navigation (anchored text, select items, etc.) should not be longer than nine (9) items. • If there is no natural ordering the first item in a list of items used for input or navigation should be the most common choice if this is meaningful. • The title of a link should not be the same as the label attached to the ‘accept’ event, (e.g. “OK”), unless the link is intended to perform the same function as the ‘accept’ event. • The title should be comprehensible if shortened to five (5) characters. • Special “device:” URLs should not be used.
Category 2 Recommendations	• Anchored text should be one or two words. • Where an image appears, it should be the only element on that line. • To allow users to place calls, use “wtai:” URLs, but also place the number to be dialed outside the link.
Test examples	• anchor1.wml, anchor2.wml, anchor3.wml

Recommendations relevant to the tag **`<card>`**:

Category 1 Recommendations	• A title should be set for each card. • The title should be comprehensible if shortened to 12 characters long. • The title should make sense without seeing the rest of the card. • The card should make sense without seeing the title. • The title should not provide the only instructional text on the card.
Test examples	Card1.wml

Recommendations relevant to the tag **`<do>`**:

Category 1 Recommendations	• If there is no natural ordering, the first item in a list of items used for input or navigation should be the most common choice of that list, if it is meaningful. • A sequence of `<do>` events should follow a consistent ordering throughout an application (e.g. 'prev', 'accept') and follow the order that they should appear on-screen. • `<do>` events should have labels. • The 'options' event should not have more than one action defined for it. • Functions that may be perceived as necessary by a user should not be accessible only by 'delete' or 'help' events. • The main task (where it exists) should not be accessible only by a `<do>` event. • A type 'accept' event should be defined for each card, perhaps to produce the `<prev>` action if there is no obvious alternative. • There should be some navigational path off the current card defined in some type of `<do>` event. • Do not rely on `<prev>` functionality to be available unless it is expressly defined. • No two `<do>` events should have the same label. • Two labels for `<do>` events, items in a select list, or anchored links that use the same text must refer to and produce the same action. • The label should be comprehensible if shortened to five (5) characters.
Category 2 Recommendations	• The word "Back" should be used as the label for an event if it has `<prev>` as its action. • The `<prev>` action should be defined for some event for each card. • The word "Back" should not be used as the label for an event unless it has 'prev' as its `<do>` event type. • `<do>` events for anything critical (e.g. back buttons) should not be defined in a `<template>`.
Test examples	do1.wml, do2.wml, do3.wml, do4.wml

Recommendations relevant to the tag **`<onevent>`**:

Category 1 Recommendations	• `onenterbackward` should not be used to move to another card by use of the `<go>` element. • `onenterforward` should not be used to go to a card where another `onenterforward` is defined to go to a card.
Test examples	onenter2.wml, onenter4.wml,

Recommendations relevant to the tag **`<optgroup>`**:

Category 1 Recommendations	• Titles should be set for all `<optgroup>`s. • A list of items used for input or navigation (anchored text, select items, etc.) should not be longer than nine (9) items.
Category 2 Recommendations	• `<optgroup>`s should be used whenever a logical breakdown of a set of options into smaller groups is useful.
Test examples	optgroup1.wml

Recommendations relevant to the tag **`<select>`**:

Category 1 Recommendations	• A select list should have a non-empty default value or `ivalue` defined for it. • A select list should be immediately preceded by at least one line of text, placing the select list in context. • A select list should have a title defined, of not more than a word or two, which makes sense when preceded by the word "edit". • `<option>`s within a single select list should not have different values for their titles. • The first five (5) characters of an `<option>` title should be sufficient to describe the purpose of the item. • A list of items used for input or navigation (anchored text, select items, etc.) should not be longer than nine (9) items. • For navigation, a list of anchored text should be used instead of select lists with `onpick` events. • The default index (`ivalue`) of a select list with `onpick` events defined should be "0", so the user is able to pick any of the options. • If a card contains a select list with `onpick` events, then that card should define `onenterbackward` and `onenterforward` actions to refresh any variable defined by the `iname` of the attribute of that select list. • `onpick` events should not be defined for select lists where multiple options can be selected. • A select list should have either all option items with `onpick` defined, or none with `onpick` defined.
Category 2 Recommendations	• Unless a dangerous action (such as authorizing a large payment) would result by selecting it, the default value of a select list should be the most common choice. This recommendation only applies if a 'most common choice' exists. • `<option>`s within a select list should not rely on the `title` attribute to indicate critical information about them (e.g. different navigation), since this information may not be displayed. • An option item like "none" should be the first item (where it makes sense) within a select list to allow no item from the list to be selected.

Test examples	select1.wml, select3.wml, select4.wml, select7.wml, select8.wml

7.2.2 Form/Entry

When the purpose of a card is to collect information from the user.

General recommendations:

Category 2 Recommendations	• Any personal information entered by users should be stored (e.g. on server, in a cookie, etc.) so that it doesn't need to be entered twice.

Recommendations relevant to the tag **`<input>`**:

Category 1 Recommendations	• An `<input>` element should have a one or two word title that makes sense when preceded by the word "edit". • Different formatting controls with identical on-screen representation should not be used in the same `<input>` element, i.e. only one of 'A' and 'X' may be used, and only one of 'a', 'x', 'M' and 'm' may be used. • Non-formatting controls (e.g. dashes) should not be used in the `format` attribute.
Category 2 Recommendations	• `password` mode should not be used for `<input>` elements unless they require only numeric input. • A web site should verify that data received from an `<input>` element matches expected format. • The text in a card immediately prior to the `<input>` element should state the number of characters to be input, if relevant, and type of information to be input, e.g. "4 digits". • The `maxlength` attribute should be used for items that have a maximum length the server will accept, but should not be relied on to provide the only checking of this length.
Test examples	input1.wml, input2.wml

Recommendations relevant to the tag **`<optgroup>`**:

Category 1 Recommendations	• Titles should be set for all `<optgroup>`s. • A list of items used for input or navigation (anchored text, select items, etc.) should not be longer than nine (9) items.
Category 2 Recommendations	• `<optgroup>`s should be used whenever a logical breakdown of a set of options into smaller groups is useful.
Test examples	optgroup1.wml

Recommendations relevant to the tag **`<select>`**:

Category 1 Recommendations	• A select list should have a non-empty default value or `ivalue` defined for it. • A select list should be immediately preceded by at least one line of text, placing the select list in context. • A select list should have a title defined, of not more than a word or two, which makes sense when preceded by the word "edit". • `<option>`s within a single select list should not have different values for their titles. • The first five (5) characters of an `<option>` title should be sufficient to describe the purpose of the item..
Category 2 Recommendations	• `<option>`s within a select list should not rely on the `title` attribute to indicate critical information about them (e.g. different navigation), since this information may not be displayed. • An option item like "none" should be the first item (where it makes sense) within a select list to allow no item from the list to be selected. • Unless a dangerous action (such as authorizing a large payment) would result by selecting it, the default value of a select list should be the most common choice. This recommendation only applies if a 'most common choice' exists.
Test examples	select1.wml

7.2.3 Appearance

How attributes on a card are rendered.

Recommendations relevant to the tag **<anchor>**:

Category 1 Recommendations	• Where anchored text appears, there should not be any other anchored text or images on that line, i.e. anchored elements are separated from other anchored elements by <br/> or <p> tags.
Category 2 Recommendations	• Where an image appears, it should be the only element on that line.
Test examples	anchor3.wml

Recommendations relevant to the tag **<card>**:

Category 1 Recommendations	• A title should be set for each card. • The title should be comprehensible if shortened to 12 characters long. • The title should make sense without seeing the rest of the card. • The card should make sense without seeing the title. • The title should not provide the only instructional text on the card. • The last item in a card should not be text if the card's order attribute is set to false. • All text immediately preceding an <input> or <select> element should be a prompt for that element if the card's order attribute is set to false. • Anchored text should not appear on a card if the card's order attribute is set to false.
Test examples	card1.wml, card2.wml

Recommendations relevant to the tag **<do>**:

Category 1 Recommendations	• If there is no natural ordering, the first item in a list of items used for input or navigation should be the most common choice of that list, if it is meaningful. • A sequence of <do> events should follow a consistent ordering throughout an application (e.g. 'prev', 'accept') and follow the order that they should appear on-screen. • <do> events of generic, experimental or vendor-specific types should not be used, e.g. avoid types such as "x-foo" or "vnd.foo". • The number of characters in labels of the 'accept' and 'options' events should not add together to greater than 15 characters if both are programmed for the same card.

	• Labels for `<do>` events should not contain spaces. • The label should be comprehensible if shortened to five (5) characters.
Category 2 Recommendations	• `<do>` events for anything critical (e.g. back buttons) should not be defined in a `<template>`.
Test examples	do1.wml, do5.wml, do6.wml

Recommendations relevant to the tag **`<fieldset>`**:

Category 1 Recommendations	• `<fieldset>`s should not be used.
Test examples	fieldset1.wml

Recommendations relevant to the tag **`<img>`**:

Category 1 Recommendations	• Images should not be wider than 81 pixels. • Image alignment should always be `bottom`. • Non-empty `alt` attributes should be set for every image. • Wireless bitmap (WBMP) images should be used instead of GIFs, Windows bitmaps, etc. • The `src` attribute of `<img>` tags should be used and set to an actual image on the server.
Category 2 Recommendations	• The `localsrc` attribute of `<img>` tags should not be used. • Images should not be higher than 44 pixels.
Test examples	Img1.wml, img2.wml

Recommendations relevant to the tag **`<input>`**:

Category 1 Recommendations	• A line break should occur immediately prior to either an `<input>` or `<select>` element.
Test examples	Misc2.wml

Recommendations relevant to the tag **`<p>`**:

Category 1 Recommendations	• Soft hyphens (`&shy;`) should not be used. • When using combinations of text alignments on a single card, each alignment should be specified (and not defaulted). • `left` (or default) text alignment should be used when a paragraph is in `nowrap` mode. • Text displayed in `nowrap` mode should be comprehensible if shortened to 10 characters long. • Non-breaking spaces (`&nbsp;`) should not be used in text.
Category 2 Recommendations	• Hyphens should not be used where the sole purpose of the hyphen is to break up a single word. • Font modes should not be relied on to convey meaning. • Words should not be longer than 12 characters in plain text and 10 characters otherwise. • Underscores should not be used in anchored text.
Test examples	p1.wml, p2.wml, p3.wml, p4.wml, p5.wml, p6.wml, p7.wml, p8.wml,

	p9.wml

Recommendations relevant to the tag **`<select>`**:

Category 1 Recommendations	- A line break should occur immediately prior to either an `<input>` or `<select>` element.
Test examples	Misc2.wml

Recommendations relevant to the tag **`<table>`**:

Category 1 Recommendations	- Anchored text should not appear in a table. - `wrap` mode should be used with tables. - A row in a table should not be more than 12 characters long.
Category 2 Recommendations	- Tables should not be used.
Test examples	Table1.wml

7.2.4 Procedure

When there is a requirement to perform tasks that do not rely on user interaction, nor need to be visible to the user.

Recommendations relevant to the tag **`<onevent>`**:

Category 1 Recommendations	• `onenterbackward` should not be used to move to another card by use of the `<go>` element. • `onenterforward` should not be used to go to a card where another `onenterforward` is defined to go to a card.
Test examples	onenter1.wml

7.3 Design Recommendations by Tag

The recommendations in this section are in the section before, and are repeated in order to provide a ready reference for developers looking up recommendations concerning particular tags.

7.3.1 <anchor>

Category 1 Recommendations	• Where anchored text appears, there should not be any other anchored text or images on that line, i.e. anchored elements are separated from other anchored elements by `<br/>` or `<p>` tags. • Anchored text should be kept as short as possible, and should not be longer than one or two words unless required for understanding. • The continuation of a list of items (anchored text, select items, etc.) should be indicated to the user – either through the use of language (e.g. “8 Items”), or with syntax through the list. • A list of items used for input or navigation (anchored text, select items, etc.) should not be longer than nine (9) items. • If there is no natural ordering the first item in a list of items used for input or navigation should be the most common choice if this is meaningful. • The title of a link should not be the same as the label attached to the ‘accept’ event, (e.g. “OK”), unless the link is intended to perform the same function as the ‘accept’ event. • The title should be comprehensible if shortened to five (5) characters. • Special “device:” URLs should not be used.
Category 2 Recommendations	• Anchored text should be one or two words. • Where an image appears, it should be the only element on that line. • To allow users to place calls, use “wtai:” URLs, but also place the number to be dialed outside the link.
Test examples	anchor1.wml, anchor2.wml, anchor3.wml

7.3.2 <card>

Category 1 Recommendations	• A title should be set for each card. • The title should be comprehensible if shortened to 12 characters long. • The title should make sense without seeing the rest of the card. • The card should make sense without seeing the title. • The title should not provide the only instructional text on the card. • The last item in a card should not be text if the card’s `order` attribute is set to `false`.

	• All text immediately preceding an `<input>` or `<select>` element should be a prompt for that element if the card's `order` attribute is set to `false`. • Anchored text should not appear on a card if the card's `order` attribute is set to `false`.
Test examples	card1.wml, card2.wml

7.3.3 <do>

Category 1 Recommendations	• If there is no natural ordering, the first item in a list of items used for input or navigation should be the most common choice of that list, if it is meaningful. • A sequence of `<do>` events should follow a consistent ordering throughout an application (e.g. 'prev', 'accept') and follow the order that they should appear on-screen. • `<do>` events should have labels. • The 'options' event should not have more than one action defined for it. • Functions that may be perceived as necessary by a user should not be accessible only by 'delete' or 'help' events. • The main task (where it exists) should not be accessible only by a `<do>` event. • A type 'accept' event should be defined for each card, perhaps to produce the `<prev>` action if there is no obvious alternative. • There should be some navigational path off the current card defined in some type of `<do>` event. • Do not rely on `<prev>` functionality to be available unless it is expressly defined. • No two `<do>` events should have the same label. • Two labels for `<do>` events, items in a select list, or anchored links that use the same text must refer to and produce the same action. • `<do>` events of generic, experimental or vendor-specific types should not be used, e.g. avoid types such as "x-foo" or "vnd.foo". • The number of characters in labels of the 'accept' and 'options' events should not add together to greater than 15 characters if both are programmed for the same card. • Labels for `<do>` events should not contain spaces. • The label should be comprehensible if shortened to five (5) characters.
Category 2 Recommendations	• The word "Back" should be used as the label for an event if it has `<prev>` as its action. • The `<prev>` action should be defined for some event for each card. • The word "Back" should not be used as the label for an event unless it

	has ‘prev’ as its `<do>` event type. • `<do>` events for anything critical (e.g. back buttons) should not be defined in a `<template>`.
Test examples	do1.wml, do2.wml, do3.wml, do4.wml, do5.wml, do6.wml

7.3.4 <fieldset>

Category 1 Recommendations	• `<fieldset>`s should not be used.
Test examples	fieldset1.wml

7.3.5 <input>

Category 1 Recommendations	• Different formatting controls with identical on-screen representation should not be used in the same `<input>` element, i.e. only one of 'A' and 'X' may be used, and only one of 'a', 'x', 'M' and 'm' may be used. • Non-formatting controls (e.g. dashes) should not be used in the `format` attribute. • An `<input>` element should have a one or two word title that makes sense when preceded by the word "edit". • A line break should occur immediately prior to either an `<input>` or `<select>` element.
Category 2 Recommendations	• `password` mode should not be used for `<input>` elements unless they require only numeric input. • The text in a card immediately prior to the `<input>` element should state the number of characters to be input, if relevant, and type of information to be input, e.g. "4 digits". • A web site should verify that data received from an `<input>` element matches expected format. • The `maxlength` attribute should be used for items that have a maximum length the server will accept, but should not be relied on to provide the only checking of this length.
Test examples	input1.wml, input2.wml, misc2.wml

7.3.6 <img>

Category 1 Recommendations	• Images should not be wider than 81 pixels. • Image alignment should always be `bottom`. • Non-empty `alt` attributes should be set for every image. • Wireless bitmap (WBMP) images should be used instead of GIFs, Windows bitmaps, etc. • The `src` attribute of `<img>` tags should be used and set to an actual image on the server.
Category 2 Recommendations	• The `localsrc` attribute of `<img>` tags should not be used. • Images should not be higher than 44 pixels.
Test examples	img1.wml, img2.wml

7.3.7 <optgroup>

Category 1 Recommendations	• Titles should be set for all `<optgroup>`s. • A list of items used for input or navigation (anchored text, select items, etc.) should not be longer than nine (9) items.
Category 2 Recommendations	• `<optgroup>`s should be used whenever a logical breakdown of a set of options into smaller groups is useful.
Test examples	optgroup1.wml

7.3.8 <onevent>

Category 1 Recommendations	• `onenterbackward` should not be used to move to another card by use of the `<go>` element. • `onenterforward` should not be used to go to a card where another `onenterforward` is defined to go to a card.
Test examples	onenter1.wml ,onenter2.wml, onenter4.wml,

7.3.9 <p>

Category 1 Recommendations	• Soft hyphens (`&shy;`) should not be used. • When using combinations of text alignments on a single card, each alignment should be specified (and not defaulted). • `left` (or default) text alignment should be used when a paragraph is in `nowrap` mode. • Text displayed in `nowrap` mode should be comprehensible if shortened to 10 characters long. • Non-breaking spaces (`&nbsp;`) should not be used in text.
Category 2 Recommendations	• Hyphens should not be used where the sole purpose of the hyphen is to break up a single word. • Font modes should not be relied on to convey meaning. • Words should not be longer than 12 characters in plain text and 10 characters otherwise. • Underscores should not be used in anchored text.
Test examples	p1.wml, p2.wml, p3.wml, p4.wml, p5.wml, p6.wml, p7.wml, p8.wml, p9.wml

7.3.10 `<select>`

Category 1 Recommendations	• A select list should have a non-empty default value or `ivalue` defined for it. • A select list should be immediately preceded by at least one line of text, placing the select list in context. • A select list should have a title defined, of not more than a word or two, which makes sense when preceded by the word “edit”. • `<option>`s within a single select list should not have different values for their titles. • The first five (5) characters of an `<option>` title should be sufficient to describe the purpose of the item. • A list of items used for input or navigation (anchored text, select items, etc.) should not be longer than nine (9) items. • For navigation, a list of anchored text should be used instead of select lists with `onpick` events. • The default index (`ivalue`) of a select list with `onpick` events defined should be “0”, so the user is able to pick any of the options. • If a card contains a select list with `onpick` events, then that card should define `onenterbackward` and `onenterforward` actions to refresh any variable defined by the `iname` of the attribute of that select list. • A line break should occur immediately prior to either an `<input>` or `<select>` element. • `onpick` events should not be defined for select lists where multiple options can be selected. • A select list should have either all option items with `onpick` defined, or none with `onpick` defined.
Category 2 Recommendations	• Unless a dangerous action (such as authorizing a large payment) would result by selecting it, the default value of a select list should be the most common choice. This recommendation only applies if a ‘most common choice’ exists. • `<option>`s within a select list should not rely on the `title` attribute to indicate critical information about them (e.g. different navigation), since this information may not be displayed. • An option item like “none” should be the first item (where it makes sense) within a select list to allow no item from the list to be selected.
Test examples	select1.wml, select3.wml, select4.wml, select7.wml, select8.wml, misc2.wml

7.3.11 `<table>`

Category 1 Recommendations	• Anchored text should not appear in a table. • `wrap` mode should be used with tables. • A row in a table should not be more than 12 characters long.
Category 2 Recommendations	• Tables should not be used.
Test examples	table1.wml

7.3.12 Miscellaneous

Category 1 Recommendations	• Different decks should have different URLs.
Category 2 Recommendations	• A collection of items should be arranged in a hierarchy so that its dimension is greater in breadth than depth. • Any personal information entered by users should be stored (e.g. on server, in a cookie, etc.) so that it doesn't need to be entered twice.
Test examples	anchor4.cgi

7.4 Example WML Decks

The following WML decks represent examples of how content may be marked up for presentation on wireless devices according to the recommendations in this style guide. These examples are not intended to show how particular applications ought to be constructed, but provide a realistic demonstration to developers of how the recommendations could be applied. The actual construction of an application is a dynamic, co-operative task between developers and their customers.

7.4.1 News Headlines

```
<wml>
  <card title="Headlines">
    <do type="accept" label="Back">
      <prev/>
    </do>
    <p>
      11 Headlines:
    </p>
    <p mode="nowrap">
      <a href="/cgi-bin/news?1">Famine in Thailand</a><br/>
      <a href="/cgi-bin/news?2">Everest climbed again</a><br/>
      <a href="/cgi-bin/news?3">Olympics rorts uncovered</a><br/>
      <a href="/cgi-bin/news?4">Price of gold plummets</a><br/>
      <a href="/cgi-bin/news?5">Silicon Valley employee scam</a><br/>
      <a href="/cgi-bin/news?6">Concerns with the Euro</a><br/>
      <a href="/cgi-bin/news?7">Sporting legend dies</a><br/>
      <a href="/cgi-bin/news?8">Elvis record sells for $$1mil</a><br/>
      <a href="#card2">3 more...</a><br/>
    </p>
  </card>
  <card id="card2" title="Headlines">
    <do type="accept" label="Back">
      <prev/>
    </do>
    <p>
      3 more headlines:
    </p>
    <p mode="nowrap">
      <a href="/cgi-bin/news?9">War in Middle East</a><br/>
      <a href="/cgi-bin/news?10">JFK conspiracy unearthed</a><br/>
      <a href="/cgi-bin/news?11">Timor border incident</a><br/>
      ---
    </p>
  </card>
</wml>
```

7.4.2 News Article

```
<wml>
  <card title="News story">
    <do type="accept" label="Back">
      <prev/>
    </do>
    <p>
      Elvis record sells for $$1mil.<br/>
      LONDON. A new record was set today when an anonymous buyer purchased
      a rare recording of Elvis songs at a Sotheby's auction.<br/>
      The recording was expected to sell for slightly more than US$$200,000
      based on an evaluation by Sotheby's staff, however the price rocketed
      up as two representatives continued to place higher and higher bids.
      <br/>
      When the Elvis tunes finally sold for $$1,025,000 it set the record
      for the most money paid for a recording of a singer involved in
      a conspiracy theory. This eclipses the $$650,000 paid last year for
      a recording of the late Marilyn Monroe gargling God Save The Queen.<br/>
      <anchor>
        Back
        <prev/>
      </anchor>
    </p>
  </card>
</wml>
```

7.4.3 Weather Forecasts

```
<wml>
  <card title="Aussie Weather">
    <do type="accept" label="Back">
      <prev/>
    </do>
    <p>
      Weather for state?
    </p>
    <p mode="nowrap">
      <img src="v.wbmp" alt="*"/><a href="#vic">Victoria,</a><br/>
      <img src="n.wbmp" alt="*"/><a href="#nsw">N. S. W.,</a><br/>
      <img src="s.wbmp" alt="*"/><a href="#sa">S. A.,</a><br/>
      <img src="q.wbmp" alt="*"/><a href="#qld">Queensland,</a><br/>
      <img src="a.wbmp" alt="*"/><a href="a.wml">A. C. T.,</a><br/>
      <img src="w.wbmp" alt="*"/><a href="#wa">W. A.,</a><br/>
      <img src="t.wbmp" alt="*"/><a href="#tas">Tasmania,</a><br/>
      <img src="nt.wbmp" alt="*"/><a href="#nt">N. T.</a><br/>
    </p>
  </card>
  <card id="vic" title="Victoria">
    <do type="accept" label="Back">
      <prev/>
    </do>
    <p>
      Which forecast?
    </p>
    <p mode="nowrap">
      <a href="v-m.wml">Metro</a><br/>
      <a href="v-s.wml">State</a><br/>
    </p>
  </card>
  <card id="nsw" title="New South Wa..">
    <do type="accept" label="Back">
      <prev/>
    </do>
    <p>
      Which forecast?
    </p>
    <p mode="nowrap">
      <a href="n-m.wml">Metro</a><br/>
      <a href="n-s.wml">State</a><br/>
    </p>
  </card>
  <card id="sa" title="South Austra..">
    <do type="accept" label="Back">
      <prev/>
    </do>
    <p>
      Which forecast?
    </p>
    <p mode="nowrap">
      <a href="s-m.wml">Metro</a><br/>
      <a href="s-s.wml">State</a><br/>
    </p>
  </card>
  <card id="qld" title="Queensland">
    <do type="accept" label="Back">
      <prev/>
    </do>
    <p>
      Which forecast?
    </p>
    <p mode="nowrap">
      <a href="q-m.wml">Metro</a><br/>
      <a href="q-s.wml">State</a><br/>
    </p>
  </card>
  <card id="wa" title="Western Aust..">
    <do type="accept" label="Back">
      <prev/>
    </do>
    <p>
      Which forecast?
    </p>
    <p mode="nowrap">
      <a href="w-m.wml">Metro</a><br/>
      <a href="w-s.wml">State</a><br/>
    </p>
  </card>
  <card id="tas" title="Tasmania">
    <do type="accept" label="Back">
      <prev/>
    </do>
    <p>
      Which forecast?
    </p>
    <p mode="nowrap">
      <a href="t-m.wml">Metro</a><br/>
      <a href="t-s.wml">State</a><br/>
    </p>
  </card>
  <card id="nt" title="Northern Ter..">
    <do type="accept" label="Back">
      <prev/>
    </do>
    <p>
      Which forecast?
    </p>
    <p mode="nowrap">
      <a href="nt-m.wml">Metro</a><br/>
      <a href="nt-s.wml">State</a><br/>
    </p>
  </card>
</wml>
```

7.4.4 Service Subscription

```
<wml>
  <card title="Join Cow Club">
    <do type="prev" label="Back">
      <prev/>
    </do>
    <do type="accept" label="Done">
      <go method="post" href="/cgi-bin/join">
        <postfield name="name" value="$name"/>
        <postfield name="age" value="$age"/>
        <postfield name="sex" value="$sex"/>
        <postfield name="address" value="$address"/>
      </go>
    </do>
    <p>
      Your name?
      <input title="Name" name="name"/>
      Your age?
      <input title="Age" name="age" format="N*N"/>
      Your gender?
      <select title="Gender" name="sex" value="m">
        <option value="m">Male</option>
        <option value="f">Female</option>
      </select>
      Your address?
      <input title="Address" name="address"/>
      <anchor>Done.
        <go method="post" href="/cgi-bin/join">
          <postfield name="name" value="$name"/>
          <postfield name="age" value="$age"/>
          <postfield name="sex" value="$sex"/>
          <postfield name="address" value="$address"/>
        </go>
      </anchor>
    </p>
  </card>
</wml>
```

7.4.5 Splash Page

```
<wml>
  <card title="Welcome" ontimer="#card2">
    <timer value="30"/>
    <do type="prev" label="Back">
      <prev/>
    </do>
    <do type="accept">
      <go href="#card2"/>
    </do>
    <p align="center">
      Welcome to<br/>
      Cow Club
    </p>
  </card>
  <card id="card2" title="Cow Club">
    <do type="accept" label="Back">
      <prev/>
    </do>
    <p>
      Cow Club tasks:
    </p>
    <p mode="nowrap">
      <img alt="*" src="join.wmbp"/><a href="join.wml">Join,</a><br/>
      <img alt="*" src="buy.wmbp"/><a href="buy.wml">Buy,</a><br/>
      <img alt="*" src="sell.wbmp"/><a href="sell.wml">Sell,</a><br/>
      <img alt="*" src="cow.wbmp"/><a href="cow.wml">Cow of the day.</a><br/>
    </p>
  </card>
</wml>
```

Annex 1 – Test Examples

The following table lists the justifications for each recommendation in relation to the test example it was derived from. The actual test examples are available separately.

Test example **anchor1.wml**

Details Combinations of anchored text of various lengths and line breaks.

Category 1 Recommendations

- Where anchored text appears, there should not be any other anchored text or images on that line, i.e. anchored elements are separated from other anchored elements by `<br/>` or `<p>` tags.

  Justification: Some browsers display each anchored element on a separate line. Anchored text designed without this rule, could be confusing because some browsers do not provide default separation between links. In addition, some browsers do not scroll the screen properly when there are many links on the same line.

- Anchored text should be kept as short as possible, and should not be longer than one or two words unless required for understanding.

  Justification: Some browsers lose the cursor when the screen is scrolled and the anchored text is large; also, some browsers do not scroll the screen properly when there are long links.

- The continuation of a list of items (anchored text, select items, etc.) should be indicated to the user – either through the use of language (e.g. “8 Items”), or with syntax through the list.

  Justification: A list of even a small number of items will flow off the screen of many browsers, so there must be some indication to the user that scrolling is necessary. While some browsers provide this indication, others do not.

- A list of items used for input or navigation (anchored text, select items, etc.) should not be longer than nine (9) items.

  Justification: The amount of scrolling required to reach all items in a single list should be minimized for ease of navigation. Some browsers’ support for select lists is inconsistent past nine items. Some browsers lose cursor context when navigating away from a list and back again, making long lists very hard to browse.

- If there is no natural ordering the first item in a list of items used for input or navigation should be the most common choice if this is meaningful.

  Justification: The most common or likely choice should require the least effort to select.

- The title of a link should not be the same as the label attached to the ‘accept’ event, (e.g. “OK”), unless the link is intended to perform the same function as the ‘accept’ event.

  Justification: Some browsers display the title of a link in place of the label for the ‘accept’ event. If these labels are not distinguishable, confusion is likely.

- The title should be comprehensible if shortened to five (5) characters.

  Justification: Some browsers display only the first 5 characters of the title of a link.

Category 2 Recommendations

- Anchored text should be one or two words.

  Justification: Short links display with greater clarity and make navigation easier. Very long links are generally problematic and may not always be able to be selected.

Test example **anchor2.wml**

Details Text anchored to special URLs such as "make call", "go to homepage", etc.

Category 1 Recommendations

- Special "device:" URLs should not be used..

  Justification: Special URLs are not supported by all browsers.

Category 2 Recommendations

- To allow users to place calls, use "wtai:" URLs, but also place the number to be dialed outside the link.

  Justification: "wtai:" URLs allow phone calls on most phones, and don't hurt on others, but some phones require the phone number to be outside a link to place a call to it.

Test example **anchor3.wml**

Details Several anchored icons on a single line.

Category 1 Recommendations

- Where anchored text appears, there should not be any other anchored text or images on that line, i.e. anchored elements are separated from other anchored elements by `<br/>` or `<p>` tags.

  Justification: Some browsers automatically place anchored text or images on new lines.

Category 2 Recommendations

- Where an image appears, it should be the only element on that line.

  Justification: Some browsers automatically place anchored images on new lines.

Test example **anchor4.cgi**

Details Generates a series of links to cards with the same main URL.

Category 1 Recommendations

- Different decks should have different URLs.

  Justification: Some browsers will load decks from the cache instead of the server if they don't have a different URL.

Test example **Card1.wml**

Details Cards with different length titles.

Category 1 Recommendations

- A title should be set for each card.

  Justification: Some browsers will display a space for the title, even when none is defined, possibly confusing the user. The title can be used to give further context to the card contents.

- The title should be comprehensible if shortened to 12 characters long..

  Justification: A long title will be truncated.

- The title should make sense without seeing the rest of the card.

  Justification: When a card is bookmarked by the user, the title may be used as the text that identifies that bookmark.

- The card should make sense without seeing the title.

  Justification: Some browsers do not display the title of the card on the screen.

- The title should not provide the only instructional text on the card.

  Justification: Some browsers do not display the title of the card on the screen.

Test example **Card2.wml**

Details Cards with `order="false"` for `<input>` elements, `<fieldset>`s and `<optgroup>`s.

Category 1 Recommendations

- The last item in a card should not be text if the card's `order` attribute is set to `false`.

  Justification: Some browsers display blank options in this case, and it is confusing.

- All text immediately preceding an `<input>` or `<select>` element should be a prompt for that element if the card's `order` attribute is set to `false`.

  Justification: Some browsers display text preceding `<input>`s or `<select>`s on a different screen, with only that `<input>` or `<select>`. Context with the original card can be lost, and become confusing.

- Anchored text should not appear on a card if the card's `order` attribute is set to `false`.

  Justification: Some browsers do not display anchored text properly on a card whose `ordered` attribute is set to `false`.

Test example **do1.wml**

Details

`<do>` event definitions in different orders.

Category 1 Recommendations

- If there is no natural ordering, the first item in a list of items used for input or navigation should be the most common choice of that list, if it is meaningful.

  Justification: Minimize key presses required for navigation.

- A sequence of `<do>` events should follow a consistent ordering throughout an application (e.g. ‘prev’, ‘accept’) and follow the order that they should appear on-screen.

  Justification: Some browsers display all do events on the screen.

Category 2 Recommendations

- `<do>` events for anything critical (e.g. back buttons) should not be defined in a `<template>`.

  Justification: Some browsers display `<template>` defined `<do>` events separate from other `<do>` events, potentially accessible only by scrolling.

Test example

do2.wml

Details

Different types of `<do>` events.

Category 1 Recommendations

- `<do>` events should have labels.

  Justification: Labels on `<do>` events provide significant assistance to the user of an application.

- The ‘options’ event should not have more than one action defined for it.

  Justification: Some browsers only allow one action to be defined for the ‘options’ event.

- Functions that may be perceived as necessary by a user should not be accessible only by ‘delete’ or ‘help’ events.

  Justification: Some browsers do not support ‘delete’ or ‘help’ events.

- The main task (where it exists) should not be accessible only by a `<do>` event.

  Justification: Access to `<do>` events on some browsers is a multiple step process which requires more key presses than alternative navigation methods.

Test example

do3.wml

Details

‘Prev’ events and actions defined in different ways. How does the behaviour of the Back key change?

Category 1 Recommendations

- A type ‘accept’ event should be defined for each card, perhaps to produce the `<prev>` action if there is no obvious alternative.

  Justification: The different browsers have different default behaviour when the accept event is not defined.

- There should be some navigational path off the current card defined in some type of `<do>` event.

  Justification: The user must have some easy way to exit the current card.

- Do not rely on `<prev>` functionality to be available unless it is expressly defined.

  Justification: Some browsers do not provide users with a "back" facility unless the `<prev>` action is defined. However, some other browsers have a hard "Back" key which cannot be programmatically overridden.

Category 2 Recommendations

- The word "Back" should be used as the label for an event if it has `<prev>` as its action.

  Justification: Some browsers apply this label as a default to an event that has 'prev' defined as its action. Always using the word 'Back" creates a consistent expectation of the behaviour.

- The `<prev>` action should be defined for some event for each card.

  Justification: Some browsers do not provide users with a "back" facility unless the `<prev>` action is defined.

- The word "Back" should not be used as the label for an event unless it has 'prev' as its `<do>` event type.

  Justification: Some browsers display the label "Back" on keys other than the normal back button, creating confusion in the user, and encouraging them to treat these other keys like a back button to cause undesirable results, e.g. tapping it multiple times and initiating actions unknowingly.

Test example

do4.wml

Details

Cards with different combinations of definitions for ‘prev’ event and `<prev>` action.

Category 1 Recommendations

- No two `<do>` events should have the same label.

  Justification: It is misleading to have two different events that are identified the same way.

- Two labels for `<do>` events, items in a select list, or anchored links that use the same text must refer to and produce the same action.

  Justification: Using the same text or label to describe an action which is offered to the user in more than one way, must always produce the same result for the user.

Test example

do5.wml

Details

Cards using unknown `<do>` types.

Category 1 Recommendations

- `<do>` events of generic, experimental or vendor-specific types should not be used, e.g. avoid types such as “x-foo” or “vnd.foo”.

  Justification: On some browsers, `<do>` events of generic, experimental or vendor-specific types are displayed as blank links.

Test example

do6.wml

Details

Cards with different length labels for `<do>` events.

Category 1 Recommendations

- The number of characters in labels of the ‘accept’ and ‘options’ events should not add together to greater than 15 characters if both are programmed for the same card.

  Justification: On some browsers, the labels of accept and option events are displayed on either side at the bottom of the screen and can interfere with each other’s readability if they are too long.

- Labels for `<do>` events should not contain spaces.

  Justification: Some browsers display the labels near the keys corresponding to these events. If spaces are present, it can look like an unlabeled key is labelled, leading to confusion.

- The label should be comprehensible if shortened to five (5) characters.

  Justification: Some browsers display only the first 5 characters.

Test example **fieldset1.wml**

Details Cards with elements grouped together using `<fieldset>`s.

Category 1 Recommendations

- `<fieldset>`s should not be used.

  Justification: Few browsers implement `<fieldset>`s. Where they are supported, `<fieldset>`s interfere with browser display of other elements.

Test example **img1.wml**

Details Different shaped images with different image alignments.

Category 1 Recommendations

- Images should not be wider than 81 pixels.

  Justification: The smallest screen width of the browser that supports images is 81 pixels, and the image cannot be scrolled horizontally.

- Image alignment should always be `bottom`.

  Justification: Other alignments may truncate the image on some browsers.

- Non-empty `alt` attributes should be set for every image.

  Justification: Not all browsers support images.

- Wireless bitmap (WBMP) images should be used instead of GIFs, Windows bitmaps, etc.

  Justification: All browsers that support images support the wireless bitmap type, but not necessarily any other image type.

Category 2 Recommendations

- Images should not be higher than 44 pixels.

  Justification: The smallest screen height of the browser that supports images is 44 pixels, although the image can be scrolled vertically.

Test example	**img2.wml**
Details	Text interspersed with images.
Category 1 Recommendations	• The `src` attribute of `<img>` tags should be used and set to an actual image on the server. **Justification:** Some browsers will refuse to display the card unless valid `src` attributes are supplied.
Category 2 Recommendations	• The `localsrc` attribute of `<img>` tags should not be used. **Justification:** Locally defined images are different for different browsers.

Test example	**input1.wml**
Details	`<input>` elements set up for `password` format.
Category 1 Recommendations	• An `<input>` element should have a one or two word title that makes sense when preceded by the word “edit”. **Justification:** The default behaviour of the browser with no title defined may lead to confusion. Providing a title that makes sense in the context of ‘Edit *Title*’, guarantees clarity across all browsers.
Category 2 Recommendations	• `password` mode should not be used for `<input>` elements unless they require only numeric input. **Justification:** Alphabetic input from a numeric keypad requires multiple presses on a single key and hiding this input from the user only compounds the difficulty of text entry.

Test example **input2.wml**

Details Text `<input>` elements with different formatting controls.

Category 1 Recommendations

- Different formatting controls with identical on-screen representation should not be used in the same `<input>` element, i.e. only one of ‘A’ and ‘X’ may be used, and only one of ‘a’, ‘x’, ‘M’ and ‘m’ may be used.

  Justification: Consistency of keying behaviour for text entry on a numeric keypad.

- Non-formatting controls (e.g. dashes) should not be used in the `format` attribute.

  Justification: Some browsers do not handle them in a way expected by users, and without warning reject the text entered by the user.

Category 2 Recommendations

- The text in a card immediately prior to the `<input>` element should state the number of characters to be input, if relevant, and type of information to be input, e.g. “4 digits”.

  Justification: This information is not always provided by the browser and in the absence of contextual knowledge of what is to be entered, provides necessary guidance, e.g. a user can believe they have entered sufficient information yet the browser remains unresponsive to their key presses.

- A web site should verify that data received from an `<input>` element matches expected format.

  Justification: Format controls do not work on every browser.

- The `maxlength` attribute should be used for items that have a maximum length the server will accept, but should not be relied on to provide the only checking of this length.

  Justification: The `maxlength` attribute provides useful assistance to the user, but cannot be guaranteed to be supported by all browsers.

Test example **misc2.wml**

Details Different input elements (`<input>` and `<select>`) with different text formatting.

Category 1 Recommendations

- A line break should occur immediately prior to either an `<input>` or `<select>` element.

  Justification: Not all browsers separate `<input>` or `<select>` elements onto different lines.

Test example **Miscellaneous – no test example appropriate**

Details n/a

Category 2 Recommendations

- A collection of items should be arranged in a hierarchy so that its dimension is greater in breadth than depth.

  Justification: Broad menus are easier to navigate than deep menus.

- Any personal information entered by users should be stored (e.g. on server, in a cookie, etc.) so that it doesn't need to be entered twice.

  Justification: Additional work for the user should be avoided.

Test example **optgroup1.wml**

Details Select lists with `<optgroup>` elements within them.

Category 1 Recommendations

- Titles should be set for all `<optgroup>`s.

  Justification: If `<optgroup>`s are supported, the relevant options are grouped and displayed under the `<optgroup>` title.

Category 2 Recommendations

- `<optgroup>`s should be used whenever a logical breakdown of a set of options into smaller groups is useful.

  Justification: `<optgroup>`s increase the structure of the information and enhance navigation. If not supported, a normal select list is displayed.

Test example **onenter1.wml**

Details Cards with `onenterforward` and `onenterbackward` defined in different ways.

Category 1 Recommendations

- `onenterbackward` should not be used to move to another card by use of the `<go>` element.

  Justification: `onenterbackward`, when used for navigation, adversely interferes with the history stack, unexpectedly trapping the user in the new card.

Category 2 Recommendations

- `onenterforward` should not be used to go to a card where another `onenterforward` is defined to go to a card.

  Justification: `onenterforward`'s events will chain if defined in this way. This creates potential for navigation to cards which cannot be predicted by the user.

Test example	**onenter2.wml**
Details	Cards that are set in an `onenterforward` loop that is entered by navigating forward.
Category 1 Recommendations	• `onenterforward` should not be used to go to a card where another `onenterforward` is defined to go to a card. [obsoletes justification from onenter1.wml] **Justification:** `onenterforward` loops may happen, locking up the browser.

Test example	**onenter4.wml**
Details	In a series of cards, `onenterbackward` is set on one card to go to another card.
Category 1 Recommendations	• `onenterbackward` should not be used to move to another card by use of the `<go>` element. [obsoletes justification from onenter1.wml] **Justification:** See above.

Test example	**p1.wml**
Details	Text using hyphens and soft-hyphens.
Category 1 Recommendations	• Soft hyphens (`&shy;`) should not be used. **Justification:** Soft hyphens are not usefully supported by most browsers.
Category 2 Recommendations	• Hyphens should not be used where the sole purpose of the hyphen is to break up a single word. **Justification:** Text that overflows will not necessarily break on a hyphen.

Test example	**p2.wml**
Details	Different length pieces of text formatted using different alignments and wrap modes.
Category 1 Recommendations	• When using combinations of text alignments on a single card, each alignment should be specified (and not defaulted). **Justification:** Different browsers default to different text alignment depending on alignments of preceding paragraphs in the card. • `left` (or default) text alignment should be used when a paragraph is in `nowrap` mode. **Justification:** Some browsers will cut off the head or tail of words

that overflow the screen boundary. `left` alignment ensures the truncated word will retain most sense.

- Text displayed in `nowrap` mode should be comprehensible if shortened to 10 characters long.

  Justification: Some browsers in 'nowrap' mode will truncate the end of text to fit the width of the screen.

Test example **p3.wml**

Details Different length pieces of text formatted using different font modes, e.g. bold, italics, etc.

Category 2 Recommendations

- Font modes should not be relied on to convey meaning.

  Justification: Different browsers support text formatting to different extents and cannot be relied on to format as may be expected.

Test example **p4.wml**

Details Lines of text with spaces in different places, some of which are long enough to wrap.

Category 21 Recommendations

- Words should not be longer than 12 characters in plain text and 10 characters otherwise.

  Justification: Consider the maximum line length to be 12 characters, including extra characters for default indentation or formatting. Words longer than this are likely to display across more than one line, making reading of text more difficult.

Test example **p5.wml**

Details Different length pieces of text on single lines, with different wrap modes.

Category 12 Recommendations

- Words should not be longer than 12 characters in plain text and 10 characters otherwise. Words should not be longer than 12 characters in plain text and 10 characters otherwise.

  Justification: See above.

Test example **p6.wml**

Details Very long pieces of text either split over several cards or displayed on a minimal number of cards.

Test example **p7.wml**

Details Anchored text containing underscores.

Category 2 Recommendations

- Underscores should not be used in anchored text.

  Justification: Some browsers underline anchored text, masking underscores.

Test example **p8.wml**

Details Different punctuation characters, and those that have special WML representation.

Category 2 Recommendations

- Underscores should not be used in anchored text.

  Justification: See above.

Test example **p9.wml**

Details Text separated by either spaces or non-breaking spaces.

Category 2 Recommendations

- Non-breaking spaces (`&nbsp;`) should not be used in text.

  Justification: Does not work on every browser. Some browsers treat 'non-breaking' spaces as 'breaking spaces'.

Test example **select1.wml**

Details Select lists with default items specified.

Category 1 Recommendations

- A select list should have a non-empty default value or `ivalue` defined for it.

  Justification: Minimize key presses and improve clarity of presentation.

- A select list should be immediately preceded by at least one line of text, placing the select list in context.

  Justification: Without additional text to place the list in context, the purpose of the list and the requirement on the user to select from it are likely to be cryptic.

- A select list should have a title defined, of not more than a word or two, which makes sense when preceded by the word "edit".

  Justification: The default behaviour of the browser with no title defined may lead to confusion. Providing a title that makes sense in the context of 'Edit *Title*', guarantees clarity across all browsers.

- `<option>`s within a single select list should not have different values for their titles.

Justification: Some browsers display each option title briefly, as the list is scrolled through. Using option titles in this way creates confusion for the user, and is not supported by all browsers.

- The first five (5) characters of an `<option>` title should be sufficient to describe the purpose of the item.

  Justification: Some browsers display only the first five characters.

Category 2 Recommendations

- Unless a dangerous action (such as authorizing a large payment) would result by selecting it, the default value of a select list should be the most common choice. This recommendation only applies if a 'most common choice' exists.

  Justification: Minimizes key presses for common navigation, and prevents the user having to enter select lists on browsers that do not initially present the full list of choices.

- `<option>`s within a select list should not rely on the `title` attribute to indicate critical information about them (e.g. different navigation), since this information may not be displayed.

  Justification: `<option>` titles are not supported by all browsers and therefore cannot be relied on for any critical display.

- An option item like "none" should be the first item (where it makes sense) within a select list to allow no item from the list to be selected.

  Justification: If a selection of "none" makes sense from a list, then that choice can only be possible if it is explicitly provided.

Test example **select3.wml**

Details Select lists of different sizes.

Category 1 Recommendations

- A list of items used for input or navigation (anchored text, select items, etc.) should not be longer than nine (9) items.

  Justification: See above.

Test example **select5.wml**

Details Select lists with `onpick` events defined.

Category 1 Recommendations

- For navigation, a list of anchored text should be used instead of select lists with `onpick` events.

  Justification: On some browsers, there is an entry screen into the select list. A list of anchored text is a more direct solution in these cases, and aids usability by reducing the number of key presses. Problems with including other elements on the same card as the select list (which will not be seen on some browsers if they come after a navigational select list) are also avoided.

- The default index (`ivalue`) of a select list with `onpick` events defined should be “0”, so the user is able to pick any of the options.

> **Justification:** In some browsers, trying to select on the default option will not trigger the `onpick` event.

- If a card contains a select list with `onpick` events, then that card should define `onenterbackward` and `onenterforward` actions to refresh any variable defined by the `iname` of the attribute of that select list.

> **Justification:** Ensures that the default index of a select list with `onpick` events will always be set to zero.

Test example **select7.wml**

Details Select lists for multiple options with `onpick` events defined.

Category 1 Recommendations

- `onpick` events should not be defined for select lists where multiple options can be selected.

> **Justification:** This is implemented differently on different browsers and its behaviour is inconsistent and generally non-sensical.

Test example **select8.wml**

Details A select list with only some option items having `onpick` defined.

Category 1 Recommendations

- A select list should have either all option items with `onpick` defined, or none with `onpick` defined.

> **Justification:** Some browsers don’t handle option items mixed with ‘onpick’ well. Also, users will be confused by the inconsistent behaviour.

Test example **table1.wml**

Details Different dimensioned tables with and without anchored text in them.

Category 1 Recommendations

- Anchored text should not appear in a table.

> **Justification:** Anchored text may destroy the format of the table.

- `wrap` mode should be used with tables.

> **Justification:** Text in `nowrap` mode within a table that overflows its cell cannot be viewed in some browsers.

- A row in a table should not be more than 12 characters long.

> **Justification:** Text which overflows may destroy the format of the table.

- A list of items used for input or navigation (anchored text, select items, etc.) should not be longer than nine (9) items.

Justification: The amount of scrolling required to reach all items in a single list should be minimized for ease of navigation. Some browsers support for select lists is inconsistent past nine items. Some browsers lose cursor context when navigating away from a list and back again, making long lists very hard to browse.

Category 2 Recommendations

- Tables should not be used.

  Justification: Tables are generally not amenable to the small screen size of most browsers.