

# Top 10 Usability Guidelines for WAP Applications:

## Steps to maximizing the user experience on devices running the Openwave™ Mobile Browser

Openwave is dedicated to understanding and enhancing the adoption of WAP applications. Over the past two years, Openwave has run monthly usability studies at its Usability and Interoperability test facilities. These studies have shown that subscribers are less likely to adopt a WAP application if its usability, or customer experience, is not optimal. The following guidelines, compiled by Openwave's Usability team, have been designed to help application developers avoid the most common usability pitfalls encountered with newly deployed WAP applications. An expanded version of these guidelines is available in the Application Style Guides on the Openwave Developer Site at <http://developer.openwave.com>.

### 1. Implement navigational menus using a <select> element

The Openwave select element implementation presents one user-friendly list with numbers which act as accelerator shortcut keys, allowing the user to select an option by pressing the corresponding number on the keypad rather than scrolling to it.

### 2. Keep soft key labels to 5 characters or less

Many devices cannot display soft key labels that exceed 5 characters, and will truncate or abbreviate any labels that are longer.

### 3. Use Wizards instead of forms

By using a single card for each input of information, the user can follow a natural process of entering information. User studies show that wizards make for a better user experience than forms. Finally, avoid having a separate 'submit' card. The last 'ok' of the wizard should in fact be the 'submit' option. This avoids an unnecessary additional click.

### 4. Keep the content that appears above select and input fields to 1 or 2 lines max (including images)

Content exceeding two lines may be truncated on smaller devices when used in conjunction with input and select fields (choice and entry cards). This may result in the loss of important information. Test your application on small-screen devices to make sure menu titles and input field prompts fit.

### 5. Assign the most commonly chosen action or most intuitive task to the accept soft key

Make it easy for the user to select the card's most common action by pressing a single key—the accept (OK) key. Don't force the user to move the cursor before pressing the accept key (e.g., by making them navigate to some text anchor first).

### 6. Don't use the <go> task to navigate to a card that is already on the history stack

If you link to a card the user has already visited (e.g., the application's main menu), be sure to navigate to the card by popping cards from the history stack instead of adding a redundant card to the history stack. This minimizes the risk of history stack overflow on memory-constrained devices, which can result in the user being unexpectedly returned to the subscriber home-page. Use the <prev> or <exit> tasks whenever navigating in the backward direction.

### 7. Allow the user to dial phone calls from the application by pressing a single key

Use the `wtai://wp/mc;` directive to allow a phone call to be invoked directly from an application via a soft key. Ensure that your application uses this feature whenever you display a phone number. Label the soft key Call so that the user can easily discover the feature.

### 8. Use the format attribute to constrain text input fields to only allow valid character types

In some applications, text entry fields can employ character format constraints. These guide the user to enter the required information. For example, if the user must enter a credit card number of 16 digits, the entry field can be formatted to accept 16 characters exactly. Other formatting is also possible; for example, the browser can be limited to accept only numeric entries. Use `format="NNNNN"` for French postal code fields.

# Top 10 Usability Guidelines for WAP Applications

## **9. Don't set a deck's expiration (maxage) to a low value unless the content is highly volatile**

Setting a short expiration time for non-volatile decks (e.g., static application menus) may force the user to reload the deck from the server even though the deck is already in the browser cache. Applications that optimize deck caching are perceived by users to be more efficient and less costly, since they minimize connection times over the wireless network. Short deck expiration values should only be set for highly volatile content (e.g., stock quotes).

## **10. Ensure all decks are smaller than 500 bytes**

The download latency for very large decks (1500–2000 bytes) can be over 10 seconds. Wireless device users perceive network latencies to be longer than they actually are, because consumer-grade devices are typically single-tasking devices (prohibiting users from performing other tasks on the device while waiting for content to load). To minimize network latencies, keep each deck as small as possible. The recommended guideline is 500 bytes or less per deck (encoded WML).

Visit the Openwave Developer Site at <http://developer.openwave.com> for the expanded Application Style Guides.

## **About Openwave**

Openwave Systems Inc. is the worldwide leader of open IP-based communication infrastructure software and applications. Openwave provides Communication Service Providers, including wireless and wireline carriers, ISPs, portals, and broadband providers worldwide, with the IP-based software infrastructure to rapidly enter the Internet economy by building boundary-free, multi-network communications services for their subscribers.

Openwave, the Openwave logo and the family of terms carrying the "Openwave" prefix are trademarks of Openwave Systems Inc. All other trademarks are the properties of their respective owners.

Copyright © 2001 Openwave Systems Inc. All Rights Reserved. February 2001.

Part Number: DSUI-R1-001

800 Chesapeake Drive  
Redwood City, CA 94063 U.S.A.  
Corporate +1 650 562 0200  
Europe +44 (0)1442 288100  
Japan +81 3 5325 9211  
<http://www.openwave.com>

