

# INTRODUCTION

## AU LANGAGE

# W.M.L.

(WAP)





## Table des matières

<b>1.0 Introduction au WAP</b> .....	4
1.1 Naissance du WAP.....	4
1.2 Quels langages pour le WAP ?.....	4
1.3 Connection d'un terminal WAP à Internet.....	4
1.4 Architecture du modèle WAP.....	4
1.5 Avenir du WAP.....	5
<b>2.0 Introduction au WML</b> .....	5
2.1 Langage WML.....	5
2.2 Construire son premier document WML.....	5
2.3 Les "cartes" en WML.....	6
2.4 Architecture d'un document WML.....	6
2.5 Règles WML.....	7
<b>3.0 Mise en page d'un document WML</b> .....	8
3.1 Paragraphes et saut de ligne.....	8
3.2 Mise en forme.....	8
<b>4.0 Les liens en WML</b> .....	10
4.1 Attributs <a>.....	10
4.2 Définition d'une "carte" (card).....	10
4.3 Liens internes (au sein d'un même document WML ("carte")).....	10
4.4 Liens internes (vers un document WML différent).....	10
4.5 Liens externe (vers un autre site WAP).....	11
4.6 La balise <anchor>.....	11
4.7 La balise <template>.....	11
<b>5.0 Les Tableaux</b> .....	12
<b>6.0 Les Boutons de navigations</b> .....	12
<b>7.0 Les Evènements en WML</b> .....	13
<b>8.0 Les Formulaires</b> .....	13
<b>9.0 Les images en WML</b> .....	14
9.1 Introduction.....	14
9.2 Réaliser une image WBMP.....	14
9.3 Insérer une image WBMP.....	14
9.3.1 Espacer l'image.....	15
9.3.2 Positionner et aligner l'image.....	15
9.3.3 Images en ROM.....	15



9.4 Conseils d'intégration.....	15
10.0 Caractères spéciaux.....	16
11.0 Caractères accentués.....	16
12.0 Caractéristiques techniques des mobiles WAP.....	17
13.0 Index des Balises par ordre alphabétiques.....	18
<A>.....	18
<ACCESS>.....	18
<ANCHOR>.....	19
<B>.....	19
<BIG>.....	19
 .....	20
<CARD>.....	20
<DO TYPE=ACCEPT>.....	21
<DO TYPE=PREV>.....	22
<DO TYPE=HELP>.....	22
<DO TYPE=RESET>.....	23
<DO TYPE=OPTIONS>.....	24
<DO TYPE=DELETE>.....	24
<DO TYPE=UNKNOWN>.....	25
<EM>.....	25
<GO HREF="URL">.....	26
<HEAD>.....	27
<I>.....	27
<IMG SRC="URL">.....	28
<INPUT TYPE=TEXT>.....	29
<INPUT TYPE=PASSWORD>.....	30
<META>.....	31
<NOOP>.....	31
<ONEVENT TYPE=ONEVENTFORWARD>.....	32
<ONEVENT TYPE=ONEVENTBACKWARD>.....	32
<ONEVENT TYPE=ONTIMER>.....	33
<ONEVENT TYPE=ONPICK>.....	33
<OPTGROUP>.....	34
<OPTION>.....	34
<P>.....	35
<POSTFIELD>.....	35
<PREV>.....	36
<REFRESH>.....	36
<SELECT>.....	37
<SETVAR>.....	38
<SMALL>.....	38
<STRONG>.....	38
<TABLE>.....	39
<TD>.....	39



---

<TEMPLATE>.....	40
<TIMER>.....	40
<TR>.....	41
<U>.....	41
<WML>.....	42
<b>14.0 Bibliographie.....</b>	<b>42</b>



## 1.0 Introduction au WAP

### 1.1 Naissance du WAP

**WAP** : Wireless application protocol (*protocole d'application par mobile*).

Né en 1999 (à l'époque, il fut présenté à l'occasion de l'Internationale Funkausstellung salon de l'électronique et des télécommunications à Berlin) et créé à l'initiative de Nokia et d'Ericson.

On peut définir le WAP comme la rencontre entre les terminaux mobiles (téléphone portable, Pocket PC,...) et Internet, qui s'est traduit par l'élaboration d'un nouveau protocole : le WAP, et par un nouveau langage : le WML. Basé sur le XML, le WML est une sorte de HTML "*light*" (afin de s'accomoder des écrans à faible résolution et du débit réduit). Il s'appuie essentiellement sur des "**cards**" (cartes) et "**decks**" (paquets de cartes).

**Note** : Le W@pforum (identique au W3C pour le HTML, XML,...) est l'organisme officiel du WAP, il décide des évolutions majeures à apporter au WAP et au langage WML, et décide des futurs standards.

### 1.2 Quels langages pour le WAP ?

Le WAP est essentiellement basé sur le WML, lui même basé sur le XML (*eXtensible Markup Language*). Le WML est donc une version très allégée du XML. Attention donc au code très strict, sensible à la case (majuscules - minuscules), balise fermante obligatoire, attributs à ne pas oublier...

Comme le HTML avec JavaScript, le WML possède son propre langage de script : WMLScript, qui même s'il est moins puissant que JavaScript reste suffisant pour le WML.

**Note** : Le PHP ou l'ASP peuvent être utilisés pour rendre vos pages WML dynamiques.

### 1.3 Connection d'un terminal WAP à Internet

Schéma indiquant le chemin emprunté par un document WML vers un terminal WAP.



### 1.4 Architecture du modèle WAP

Comme Internet qui dispose de ses protocoles (*HTTP, TCP/IP...*), le WAP est construit autour d'une architecture similaire et donc de protocoles spécifiques, adaptés et optimisés pour les liaisons sans fil.



Ci-dessous les différents protocoles utilisés par le WAP.

**a ) WAE - Wireless Application environment**

Fournit un environnement de travail (WML, WMLScript, URI).

**b ) WSP - Wireless Session Protocol**

Fournit l'interface qui gère les sessions.

**c ) WTP - Wireless Transport Protocol**

Transporte l'information (identique au protocole HTTP sur Internet).

**d ) WDP - Wireless Datagram Protocol**

Fiabilise les différentes transactions (identique au protocole TCP/IP sur Internet).

**e ) WTLS - Wireless Transport Layer Protocol**

Gère les standards industriels de sécurité (identique à SSL sur Internet).

## 1.5 Avenir du WAP

Il est clair que l'on se trouve à l'ère 0 du WAP... C'est surtout un pas vers les services de demain. En effet, vu l'écran réduit et monochrome ainsi que la bande passante très faible, le WAP est très délicat à aborder pour espérer un quelconque rendement et très limité pour des développements "lourds".

On attend donc avec impatience le GPRS, EDGE et surtout l'UMTS pour obtenir des services interactifs (multimédia) et démarrer le m-commerce (Mobile Commerce).

	Débit (Kbps)	Année
<b>GSM</b>	9,6	1992
<b>GSM II</b>	14,4	1997
<b>GPRS</b>	9 à 170	2001
<b>EDGE</b>	300	2001-2002
<b>UMTS</b>	144 à 344	2002-2003

## 2.0 Introduction au WML

### 2.1 Langage WML

**WML** : Wireless Markup Language.

Basé sur le XML, le WML est tout de même très proche du HTML (syntaxe).

Attention tout de même à respecter les spécifications très sévère.

### 2.2 Construire son premier document WML

Pour construire un document au format W.M.L., il faut posséder un éditeur de texte quelconque tel que que Word, Wordpad, le Bloc-note de Windows...en veillant à enregistrer votre document au format WML. Comme vous allez le voir, le nombre de balises W.M.L., peu élevé, n'excède pas quarante.



## 2.3 Les " cartes " en WML

Le WML s'appuie essentiellement sur la notion de " carte ". Les documents WML sont nommés "decks" (paquet de cartes). Chaque deck est quand à lui composé de plusieurs " cards " (cartes).

**Note :** On peut dire que la notion de "carte" en WML (balise <card>) est similaire à la notion d'ancre en HTML (balise <A NAME>). On peut, donc, avec un seul document WML, créer un site WAP qui possède plusieurs sections.

*Exemple :*

```
<wml>
  <card>
    ...
    ...
  </card>
</wml>
```

## 2.4 Architecture d'un document WML

On peut découper un document WML en plusieurs parties et ainsi analyser la structure et l'architecture d'une page WAP classique.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="carte1">
    <p>ALL HTML - Langage WML</p>
    <select>
      <option onpick="#1">Carte 1</option>
      <option onpick="#2">Carte 2</option>
    </select>
  </card>
  <card id="carte2">
    <b>ALL HTML - Langage HDML</b>
  </card>
</wml>
```

### **a ) Entête**

Déclaration de l'entête spécifiant le type de version WML utilisé, ici version 1.1.

### **b ) Définition de la première carte**

Déclaration de la première carte par le biais de la balise <card id="...">

### **c ) Contenu de la première carte**

Exemple de contenu d'un document WML. Ici la balise <select>.

### **d ) Définition de la deuxième carte**

Déclaration de la deuxième carte.



### e ) Fin du document

Fin du document par le biais de la braise fermante `</wml>`.

## 2.5 Règles WML

Vu que le WML est avant tout basé sur le XML, il est donc aussi sévère sur l'interprétation du code, ce qui est avant tout une bonne chose pour les développeurs.

Voici quelques règles à respecter scrupuleusement :

**a ) *Minuscules*** : Les balises, attributs et valeurs doivent toujours être spécifiées en minuscules.

**b ) *Attributs*** : Attention certains attributs sont obligatoires.

**c ) *Guillemet*** : N'oubliez surtout pas d'entourer les valeurs des attributs par des guillemets.

**d ) *Attribut ALT*** : Certains mobiles ne savent pas lire les images au format WBMP, il est donc de rigueur de fournir une alternative avec l'attribut ALT.

**e ) *Entête*** : L'entête d'un document WML est obligatoire. Ces premières lignes définissent la déclaration de normes XML (car le WML est dérivé de la version 1.0 du XML).

**f ) *id*** : Evitez de dépasser 8 caractères pour l'ID à l'intérieur de la balise `<card>` (certains mobiles peuvent poser problème).

**g ) *Case*** : Attention le WML est sensible à la case. Par exemple valeur1 est différent de Valeur1 et de vaLeur1.

**h ) *Ordre*** : Bien suivre l'ordre des fermeture/ouverture.

**i ) *Fermeture*** : N'oubliez pas de fermer correctement toutes vos balises !!

Si une de ces recommandations n'est pas prise en compte, vous obtiendrez indéfiniment des erreurs lors de la visualisation de vos documents sur n'importe quels terminaux.

N'hésitez donc pas à utiliser une vérification de code WML (disponible dans tous les éditeurs WML), et à vérifier grandeur nature vos pages WML.



## 3.0 Mise en page d'un document WML

### 3.1 Paragraphes et saut de ligne

Même s'il est difficile de réaliser un document WML aéré, principalement du à l'écran réduit du terminal, de nombreuses balises sont présentes (dans la plupart des cas les mêmes que pour le HTML) afin d'obtenir une présentation lisible.

#### **a ) Paragraphe**

La balise `<p>` exprime le début d'un paragraphe, c'est à dire un double espace. Elle doit impérativement être suivie de sa balise de fermeture `</p>` .

La balise `<p>` possède l'attribut *align* qui sert à aligner le texte et prend 4 valeurs :

1. *left* : Aligne le texte sur la marge de gauche.
2. *right* : Aligne le texte sur la marge de droite.
3. *center* : Centre le texte entre les marges.

**Note** : L'attribut mode possède deux valeurs : *wrap* et *nowrap*. la valeur *nowrap* permet de ne pas effectuer de saut à la ligne (la valeur *wrap* permet l'inverse et est la valeur par défaut). Attention à ne pas masquer le texte (*mode=nowrap*) suivant la largeur de l'écran du terminal.

*Exemple* :

```
Ceci est un exemple  
<p align=center>  
avec la balise p  
</p>
```

#### **b ) Saut de ligne**

La balise `<br>` effectue un retour à la ligne (simple espace) identique à la touche Entrée de votre clavier (sous traitement de texte). Elle ne possède pas d'attribut et contient elle même sa "propre fermeture" (`/`).

*Exemple:*

```
Ceci est un  
<br/>  
exemple  
<br/>  
avec la balise br
```

### 3.2 Mise en forme

Il existe très peu de balises dites de mise en forme, dû principalement au navigateur WAP installé sur les terminaux mobiles, et l'écran dépourvu de couleurs.

**Note** : Certaines de ses balises ne seront peut-être pas reconnues sur tous les navigateurs WAP.



### a ) Augmenter la taille des caractères

Elle se fait uniquement avec les balises `<big>` et `</big>`

**Note** : Vous ne pouvez pas jouer sur le facteur d'agrandissement du texte.

*Exemple* :

```
<big>ALL HTML</big>
```

### b ) Réduire la taille des caractères

Elle se fait uniquement avec les balises `<small>` et `</small>`

**Note** : Vous ne pouvez pas jouer sur le facteur de réduction du texte.

*Exemple* :

```
<small>ALL HTML</small>
```

### c ) Gras

Pour mettre en gras du texte on optera pour la balise `<b>` (ou `<strong>`)

*Exemple* :

```
<b>ALL HTML</b>
```

### d ) Italique

Pour mettre en italique du texte on optera pour la balise `<i>` (ou `<em>`)

*Exemple* :

```
<i>ALL HTML</i>
```

### e ) Souligner

Pour souligner du texte on optera pour la balise `<u>`

**Note** : Souligner du texte peut être confondu avec un lien.

*Exemple* :

```
<u>ALL HTML</u>
```



## 4.0 Les liens en WML

### 4.1 Attributs de <a>

La balise <a> permet d'établir un lien hypertexte, à l'intérieur même de votre document WML vers un autre document, ou encore vers l'extérieur (vers un autre site WAP). La structure d'un lien dans un document WML est identique au HTML :

```
<a href="...">Lien</a>
```

Ses principaux attributs sont :

- *accesskey* : Permet de spécifier une touche (0 à 9) de raccourci.
- *href* : Spécifie le type de lien, interne ou externe.
- *class* : Nom de la classe donnée à la balise à laquelle elle appartient.
- *title* : Permet de spécifier une description du lien (identique à l'attribut ALT pour une image).

**Note** : La fermeture de la balise <a> est obligatoire : </a>

### 4.2 Définition d'une "carte" (card)

Pour définir une "carte" (card) il suffit tout simplement d'utiliser la balise <card>

ex :

```
<card id="section1">
```

**Note** : Bien entendu on peut définir plusieurs "cartes" à l'intérieur d'un même document WML.

### 4.3 Liens internes (au sein d'un même document WML ("carte"))

ex :

```
<a href="#section1">Section 1</a>
```

**Note** : Attention à ne pas oublier le # devant l'ID.

Dans cette exemple lors d'un clic sur ce lien, vous serez amené vers la "carte" (card) appelée section1.

### 4.4. Liens internes (vers un document WML différent)

a ) Lien vers un document WML situé dans le même répertoire

ex :

```
<a href="page2.wml">Page 2</a>
```

b ) Lien vers un document WML situé dans un répertoire différent

ex :

```
<a href="repertoire/page2.wml">Page 2</a>
```



c ) Lien vers un document WML situé dans le même répertoire et vers une "carte"  
ex :

```
<a href="page2.wml#section1">Page 2</a>
```

d ) Lien vers un document WML situé dans un répertoire différent et vers une "carte"  
ex :

```
<a href="repertoire/page2.wml#section1">Page 2</a>
```

### 4.5 Liens externe (vers un autre site WAP)

ex :

```
<a href="http://www.monsite.com/mapage.wml">Ma page</a>
```

### 4.6 La balise <anchor>

A l'aide de la balise **<anchor>**, on peut créer également des liens. Cette balise peut être couplée avec :

- La balise **<go>** : Pour créer un lien ou exécuter un programme CGI.
- La balise **<refresh>** : Pour recharger le document en cours.
- La balise **<prev>** : Pour réaliser un retour vers la page précédente.

a) Balise **<anchor>** couplée avec la balise **<go>**

ex :

```
<anchor>Section1<go href="#section1"/></anchor>
```

b) Balise **<anchor>** couplée avec la balise **<refresh>**

ex :

```
<anchor>Actualiser</refresh></anchor>
```

c) Balise **<anchor>** couplée avec la balise **<prev>**

ex :

```
<anchor>Précédente</prev></anchor>
```

### 4.7 La balise <template>

La balise template permet de créer un lien accessible sur toutes les "cartes" d'un document WML. Elle doit être combinée avec la balise **<do>**

```
<template>  
<do type="prev" label="Précédente">  
<prev/>  
</do>  
</template>
```



Dans cet exemple, un lien vers la page précédente sera actif sur toutes les cartes d'un document WML.

**Note** : Cette balise doit être définie avant les "cartes".

## 5.0 Les Tableaux

La structure de base d'un tableau est la suivante :  
(exemple d'un tableau d'une ligne (*tr*) de deux cellules(*td*))

```
<table>
  <tr>
    <td>Cellule numéro 1</td>
    <td>Cellule numéro 2</td>
  </tr>
</table>
```

Ses principaux attributs sont :

- `<table>...</table>`: Définition d'un tableau.
- `<table id="mon tableau">` : Définition du nom du tableau.
- `<table align="center">` : Définition de l'alignement du tableau (left, right, center).
- `<table title="titre">` : Définition du titre du tableau.
- `<table columns="7">` : Définition du nombre de colonnes du tableau.
- `<tr>...</tr>` : Définition d'une ligne du tableau.
- `<tr id="nom de la ligne">` : Définition du nom d'une ligne.
- `<td>...</td>` : Définition d'une cellule à l'intérieur d'une ligne.
- `<td id="nom de la cellule">` : Définition du nom d'une cellule.

## 6.0 Les Boutons de navigations

La définition d'un bouton est donnée par la syntaxe suivante :

```
<do></do>
```

**Note** : par défaut retourne à la card initiale, celle qui figure en premier dans le code.

Ses principaux attributs sont :

- `<do label="intitulé"></do>` : Définition de l'intitulé du bouton.
- `<do type="prev" label="retour"><prev/></do>` : Définition d'un bouton de retour vers la page précédente.
- `<do type="accept" label="retour vers une card"><go href=#nom de card/></do>` : Définit un bouton de retour vers une card précise.
- `<do type="reset" label="annulation"><reset/></do>` : Annule le contenu d'un formulaire.
- `<do type="delete" label="remise à zéro"><delete/></do>` : Remise à zéro du contenu d'un formulaire.



## 7.0 Les Evènements en WML

En WML, un évènement (*event*) survient lorsqu'un utilisateur accède à une carte, sélectionne des options, ou lorsqu'un timer arrive à terme. Ces évènements déclenchent une action.

Les actions qui s'appliquent à toutes les cartes d'un deck peuvent être définies dans un élément *template*. Les cartes peuvent disposer d'actions supplémentaires qui priment sur les précédentes. C'est le cas pour les balises **onevent** du même type et pour les éléments **do** avec le même attribut *name*.

L'action *noop* est conçue comme un mode de désactivation d'actions pour une carte spécifique.

Une fois la durée écoulée, une action *ontimer* est déclenchée au sein d'un *template* ou d'une carte tant que l'utilisateur ne quitte pas la carte ou ne déclenche pas une autre action.

## 8.0 Les Formulaires

Les listes de sélection sont définies par la balise "*select*", chaque option dans la liste étant introduite par la balise "*option*". De même que pour "*input*", l'attribut "*name*" de la balise "*select*" doit contenir le nom d'une variable, l'attribut "*title*" spécifie le titre de la page intermédiaire générée par le terminal et l'attribut booléen "*multiple*" autorise ou interdit le choix multiple.

L'attribut principal de la balise "*option*" est "*value*" et contient la valeur qui sera stockée dans la variable résultat.

Définition d'une zone de texte :

```
<input type="texte"/>
```

Définition du nom d'une zone de texte :

```
<input type="texte" id="nom de la zone">
```

Définition du nom de la variable attribué à cette zone :

```
<input type="texte" name ="nom de la variable">
```

Définition de l'envoi du contenu à un mail :

*name* correspond au nom de la zone (*id*)

*value* correspond au nom de la variable associé à la zone (*name*)

```
<anchor>
  Envoyer le formulaire
  <go method="post" href="http://www.serveur.com/page.wml">
    <postfield name="zone1" value="$ (zone1) ">
    <postfield name="zone2" value="$ (zone2) ">
  </go>
</anchor>
```



## 9.0 Les images en WML

### 9.1 Introduction

**WBMP** : Wireless BitMaP.

Le WML possède son propre format d'image : Le WBMP. Ses principales caractéristiques (adaptées bien entendu aux écrans à faible résolution et au débit réduit) :

- 2 couleurs (noir et blanc) (adapté du format BMP)
- 1 bit - bichromatique
- TYPE 0
- Pas de compression possible
- Suit les recommandations du WAP - WAE

### 9.2 Réaliser une image WBMP

Il existe plusieurs façons de créer une image au format WBMP, où plutôt de transformer cette image au format WBMP.

**a )** Un logiciel graphique classique comme PhotoShop, ou Paint Shop Pro, par l'intermédiaire d'un plug-in (WBMP Plugin).

**b )** Des outils autorisant la conversion de fichiers image au format GIF ou JPEG vers le format WBMP.

**c )** Un service Web qui convertit en ligne ("on line") images au format GIF ou JPEG vers le format WBMP.

### 9.3 Insérer une image WBMP

Le protocole pour intégrer une image WBMP à une page WML est identique à celui du HTML. On se sert donc de la balise `<img>`

Cette balise a de nombreux attributs, les plus importants étant bien entendu *ALT* (qui spécifie une alternative à l'image - obligatoire !) et *SRC* qui permet de spécifier l'URL complète (chemin) du fichier :

```

```

Le nom de fichier doit être remplacé par le nom de l'image et son extension (image.wbmp). Si votre fichier image.wbmp se trouve dans un autre répertoire que vos pages, par exemple dans un répertoire img, cela donne :

```

```

Vous pouvez préciser la taille de l'image au navigateur WAP, celui-ci chargera plus rapidement votre page. En effet, il n'a pas à la recalculer et, si jamais le navigateur



du mobile ne peut afficher les images WBMP, la mise en page ne sera pas perturbée.

Ces dimensions correspondent à la taille en pixels de l'image. Ici, l'image fait 20 pixels de large sur 20 pixels de hauteur.

```

```

### 9.3.1 Espacer l'image

Avec les attributs *hspace=n* *vspace=n* (n étant un nombre en pixels), vous pouvez fixer un espace vertical et horizontal autour de votre image.

```

```

### 9.3.2 Positionner et aligner une image

Par défaut, l'image est placée sur une ligne et le texte, s'il est présent, est aligné sur le bas de l'image. Si vous voulez encadrer une image avec du texte (style journal) l'attribut *align* est requis. il possède 3 attributs:

- *middle* : Texte au milieu.
- *bottom* : Texte en bas.
- *top* : En haut.

```

```

### 9.3.3 Images en ROM

A l'aide de l'attribut *localsrc*, vous pouvez afficher des dessins qui sont contenus dans la rom (mémoire morte) du terminal (suivant les modèles).

```

```

## 9.4 Conseils d'intégration

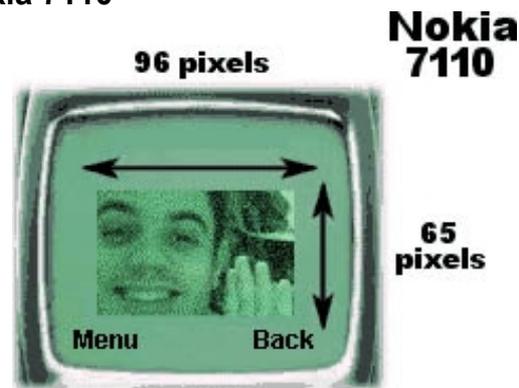
**a )** N'oubliez pas le *alt* (attribut obligatoire !), qui permet de fournir une représentation contextuelle aux terminaux WAP non-graphiques.

**b )** Vos images doivent avoir un poids maximum de 0.5 Ko soit 500 octets (sinon vous risquez de saturer la mémoire du terminal), vous pouvez en théorie aller jusqu'à 1500 octets!!!

**c )** Attention à ne pas dépasser certaines hauteurs (*height*) et largeurs (*width*), donc respectez les écrans peu larges des terminaux mobiles.



## Exemple avec un Nokia 7110



### 10.0 Caractères spéciaux

Voici une liste des caractères spéciaux. Il suffit d'insérer le code du caractère pour l'obtenir dans votre page, ainsi vous garantissez que ces caractères spéciaux seront reconnus correctement par tous les terminaux mobiles (téléphone portable, Pocket PC,...).

Caractère	Code	Signification
<	&lt;	Inférieur à
>	&gt;	Supérieur à
'	&apos;	Apostrophe
"	&quot;	Guillemets
&	&amp;	Et (commercial)
\$	\$\$	Dollar
	&nbsp;	Espace insécable
-	&shy;	Trait d'union

### 11.0 Caractères accentués

Voici une liste des caractères accentués (langue française). Il suffit d'insérer le code du caractère pour l'obtenir dans votre page.

**Note :** Afin de garantir une compatibilité maximum, il est préférable d'omettre ces caractères accentués spécifiques à la langue française.

Caractère	Code	Signification
à	&#xE0;	a accent grave
â	&#xE2;	a accent circonflexe
é	&#xE9;	e accent aigu
è	&#xE8;	e accent grave
ê	&#xEA;	e accent circonflexe
ô	&#xF4;	o accent circonflexe
ù	&#xF9;	u accent grave
û	&#xFB;	u accent circonflexe
ç	&#xE7;	c cédille
©	&#xA9;	Copyright
®	&#xAE;	Marque déposée



## 12.0 Caractéristiques techniques des mobiles WAP

Voici une liste des différents téléphones compatibles WAP, ainsi que leurs caractéristiques techniques (lignes, pixels).

**Note** : Tous ces modèles sont compatibles avec la version **WAP 1.1**.

<b>Mobiles</b>	<b>Nbr de lignes</b>	<b>Caractères / Lignes</b>	<b>Pixels (W x H)</b>
Alcatel (One touch 301 à 304)	5	12	90 x 45
Alcatel (One touch 501)	8	12	-
Alcatel (One touch 701)	8	12	-
Alcatel (One touch view DB)	2	12	90 x 154
Alcatel (OTV db WAP)	4	12	-
Benefon Q	6	-	-
Bosch 1886	5	-	-
Bosch 820	5	-	-
Ericsson (R380)	5	-	360 x 120
Ericsson (R320S)	5	12	96 x 65
Ericsson (R250S)	3	12	-
Kenwood ED 628	5	16	-
Motorola (T2288)	5	16	96 x 64
Motorola (V2288)	5	16	96 x 64
Motorola (P7389)	5	16	96 x 64
Nec (DB 6000)	-	-	120 x 160
Nokia (9110i)	5	13	640 x 200
Nokia (7110)	6	13	96 x 65
Nokia (6210)	5	13	96 x 60
Nokia (6250)	5	13	96 x 60
Panasonic (GD 93)	6	16	-
Sagem (MW939)	8	12	-
Samsung (SGH-A110)	6	12	128 x 64
Samsung (SGH-2400)	4	12	-
Samsung (SGH-2410)	4	12	-
Samsung (SGH-3500)	4	12	96 x 35
Siemens (C35i)	5	16	-
Siemens (M35i)	5	16	-
Siemens (S35i)	7	16	-
Sony (CMD-Z5)	6	16	96 x 72
Trium (Cosmo @)	4	13	-
Trium (Geo-@)	6	15	-
Trium (Mundo)	-	-	-
Trium (XS)	5	15	-



## 13.0 Index des Balises par ordre alphabétiques

### Balise <A>

#### 1. Description

Cette balise permet de définir un lien. Le lien est une URL ou une URI et se place dans l'attribut href. On peut utiliser le caractère # pour réaliser un lien local au fichier. Le texte se trouvant entre la balise <A> et </A> sera l'affichage donné au lien.

#### 2. Attributs

<b>href</b>	Renseigne l'URL ou URI du lien
Exemple	href="http://www.serveur.com/page.wml "
<b>title</b>	Renseigne le label attribué au lien (optionnel).
Exemple	title="lien1"

#### 3. Exemple

```
<A href="http://www.serveur.com/page.wml" title="lien1">Ma page WAP</A>
```

#### 4. Remarque

On peut trouver entre les balises <A> et </A> les balises suivantes : <BR>, <IMG> et du texte.

### Balise <ACCESS>

#### 1. Description

Cette balise est définie dans une balise <HEAD>. Elle permet de spécifier des contrôles d'accès au niveau du domaine et du répertoire. Lorsque l'utilisateur navigue de card en card et qu'il rencontre une balise access, la card est comparée au nom de domaine et au répertoire de provenance. Si cela correspond, alors la nouvelle card est lue.

#### 2. Attributs

<b>domain</b>	Renseigne l'URL d'un nom de domaine, par défaut le nom de domaine de la card.
Exemple	domain="serveur.com"
<b>path</b>	Renseigne un répertoire relatif ou un chemin complet. Par défaut, la valeur est "/"
Exemple	path="/wap/"

#### 3. Exemple

```
<HEAD>  
  <ACCESS domain="serveur.com" path="/wap/">  
</HEAD>
```

#### 4. Remarque

La balise ACCESS se trouve obligatoirement dans une balise <HEAD>.



## **Balise <ANCHOR>**

### 1. Description

Cette balise est utilisée pour la gestion d'évènements. Elle permet de distinguer un contenu ancré d'un contenu simple. Par exemple, en HTML, une ancre est utilisée pour souligner et afficher dans une couleur différente d'un contenu normal.

### 2. Attributs

**title** Renseigne un nom à l'ancre  
Exemple TITLE="Ancre1"

### 3. Exemple

```
<ANCHOR TITLE="Ancre1">Ma premier site WAP</ANCHOR>
```

### 4. Remarque

Entre les balises <ANCHOR> et </ANCHOR>, on peut trouver les balises suivantes : <br>, <go>, <img>, <prev>, <refresh> et du texte.

## **Balise <B>**

### 1. Description

Cette balise est utilisée pour formater du texte en caractère gras. Le texte se trouvant entre la balise <B> et </B> sera en caractère gras.

### 2. Attributs

Pas d'attributs.

### 3. Exemple

```
<B>Mon premier site WAP</B>
```

## **Balise <BIG>**

### 1. Description

Cette balise est utilisée pour formater du texte en caractère gros. Le texte se trouvant entre la balise <BIG> et </BIG> sera en caractère large.

### 2. Attributs

Pas d'attributs.

### 3. Exemple

```
<BIG>Mon premier site WAP</BIG>
```



## **Balise <BR/>**

### 1. Description

Cette balise permet de passer à la ligne. Il n'y a pas de balises d'ouvertures et de fermetures. Une seule balise : `<BR/>`.

### 2. Attributs

Pas d'attributs.

### 3. Exemple

`<BR/>`

## **Balise <CARD>**

### 1. Description

Le code WML doit être présent dans une card. Une card représente un écran d'affichage de mobile. Elle peut contenir une combinaison d'information formatée, du contenu affichable, des instructions... Chaque card doit contenir un ou plusieurs éléments.

### 2. Attributs

<b><i>id</i></b>	Renseigne un identificateur à la card.
Exemple	ID="card1"
<b><i>newcontext</i></b>	Renseigne si la card lue doit reprendre le contexte de la précédente (true false). SI l'attribut est positionné à TRUE, toutes les variables contextuelles sont nétoyées, l'historique est effacé.
Exemple	NEWCONTEXT="true  <b>false</b> "
<b><i>onenterbackward</i></b>	Renseigne une URL à lancer si la card est exécutée à partir d'une balise <code>&lt;PREV&gt;</code> .
Exemple	ONENTERBACKWARD="URL"
<b><i>onenterforward</i></b>	Renseigne une URL à lancer si la card est exécutée à partir d'une balise <code>&lt;GO&gt;</code> .
Exemple	ONENTERFORWARD="URL"
<b><i>ontimer</i></b>	Renseigne une URL à lancer si une balise <code>&lt;TIMER&gt;</code> est exécutée.
Exemple	ONTIMER="URL"
<b><i>ordered</i></b>	Indique si le contenu doit être présenté sous forme de liste ou écran/écran lors de saisies par exemple.
Exemple	ORDERED=" <b>true</b>  false"
<b><i>title</i></b>	Renseigne le titre de la card. Le titre sera affiché en haut de l'écran du mobile.
Exemple	TITLE="Mon site WAP"



### 3. Exemple

```
<CARD ID="1" TITLE="dumobile" NEWCONTEXT="TRUE">
```

Contenu ici

```
</CARD>
```

### 4. Remarque

Un fichier WML peut contenir plusieurs card.

Entre la balise `<CARD>` et `</CARD>` on peut avoir les balises suivantes :

`<ONEVENT>`, `<DO>`, `<P>` et `<TIMER>`.

## **Balise <DO TYPE=ACCEPT>**

### 1. Description

L'instruction se trouvant entre les balises `<DO TYPE=ACCEPT>` et `</DO>` sera exécuté sur une validation de l'utilisateur.

### 2. Attributs

<b>type</b>	Renseigne une action prédéfinie.
Exemple	TYPE="ACCEPT"
<b>label</b>	Renseigne un label identifiant l'évènement. Ce label doit se restreindre à 6 caractères max (WML 1.1).
Exemple	LABEL="Go!"
<b>name</b>	Renseigne un nom à l'évènement.
Exemple	NAME="Lancer"
<b>optional</b>	Indique à l'utilisateur qu'il peut ignorer certains éléments.
Exemple	OPTIONAL="true   false"

### 3. Exemple

```
<DO TYPE="ACCEPT" LABEL="Go!" OPTIONAL="false">  
  <GO HREF="#startcard"/>  
</DO>
```

### 4. Remarque

Entre la balise `<DO TYPE=ACCEPT>` et `</DO>`, on peut retrouver les balises suivantes :

`<GO>` ou `<NOOP>` ou `<PREV>` ou `<REFRESH>`



## **Balise <DO TYPE=PREV>**

### 1. Description

L'instruction se trouvant entre les balises <DO TYPE=PREV> et </DO> sera exécuté sur un retour sur une card précédente actionnée par l'utilisateur.

### 2. Attributs

<b>type</b>	Renseigne une action prédéfinie.
Exemple	TYPE="PREV"
<b>label</b>	Renseigne un label identifiant l'évènement. Ce label doit se restreindre à 6 caractères max (WML 1.1).
Exemple	LABEL="Go!"
<b>name</b>	Renseigne un nom à l'évènement.
Exemple	NAME="Lancer"
<b>optional</b>	Indique à l'utilisateur qu'il peut ignorer certains éléments.
Exemple	OPTIONAL="true   <b>false</b> "

### 3. Exemple

```
<DO TYPE="PREV" LABEL="Previous">  
  <GO HREF="#previouscard"/>  
</DO>
```

### 4. Remarque

Entre la balise <DO TYPE=PREV> et </DO>, on peut retrouver les balises suivantes : <GO> ou <NOOP> ou <PREV> ou <REFRESH>

## **Balise <DO TYPE=HELP>**

### 1. Description

L'instruction se trouvant entre les balises <DO TYPE=HELP> et </DO> sera exécuté sur une demande d'aide de l'utilisateur sur une card.

### 2. Attributs

<b>type</b>	Renseigne une action prédéfinie.
Exemple	TYPE="HELP"
<b>label</b>	Renseigne un label identifiant l'évènement. Ce label doit se restreindre à 6 caractères max (WML 1.1).
Exemple	LABEL="Help"
<b>name</b>	Renseigne un nom à l'évènement.
Exemple	NAME="Aide"
<b>optional</b>	Indique à l'utilisateur qu'il peut ignorer certains éléments.
Exemple	OPTIONAL="true   <b>false</b> "



## 3. Exemple

```
<DO TYPE="HELP" LABEL="Help">  
  <GO HREF="#helpcard"/>  
</DO>
```

## 4. Remarque

Entre la balise `<DO TYPE=HELP>` et `</DO>`, on peut retrouver es balises suivantes : `<GO>` ou `<NOOP>` ou `<PREV>` ou `<REFRESH>`

## **Balise <DO TYPE=RESET>**

### 1. Description

L'instruction se trouvant entre les balises `<DO TYPE=RESET>` et `</DO>` sera exécuté sur une demande de réinitialisation du contexte.

### 2. Attributs

<b>type</b>	Renseigne une action prédéfinie.
Exemple	TYPE="RESET"
<b>label</b>	Renseigne un label identifiant l'évènement. Ce label doit se restreindre à 6 caractères max (WML 1.1).
Exemple	LABEL="Reset"
<b>name</b>	Renseigne un nom à l'évènement.
Exemple	NAME="Reset"
<b>optional</b>	Indique à l'utilisateur qu'il peut ignorer certains éléments.
Exemple	OPTIONAL="true   false"

### 3. Exemple

```
<DO TYPE="RESET" LABEL="Reset">  
  <GO HREF="#resetcard"/>  
</DO>
```

### 4. Remarque

Entre la balise `<DO TYPE=RESET>` et `</DO>`, on peut retrouver les balises suivantes : `<GO>` ou `<NOOP>` ou `<PREV>` ou `<REFRESH>`



## **Balise <DO TYPE=OPTIONS>**

### 1. Description

L'instruction se trouvant entre les balises <DO TYPE=OPTIONS> et </DO> sera exécuté sur une opération ajoutée au menu Options.

### 2. Attributs

<b>type</b>	Renseigne une action prédéfinie.
Exemple	TYPE="OPTIONS"
<b>label</b>	Renseigne un label identifiant l'évènement. Ce label doit se restreindre à 6 caractères max (WML 1.1).
Exemple	LABEL="Help"
<b>name</b>	Renseigne un nom à l'évènement.
Exemple	NAME="Aide"
<b>optional</b>	Indique à l'utilisateur qu'il peut ignorer certains éléments.
Exemple	OPTIONAL="true   false"

### 3. Exemple

```
<DO TYPE="OPTIONS" LABEL="Help">  
  <GO HREF="#helpcard"/>  
</DO>
```

### 4. Remarque

Entre la balise <DO TYPE=OPTIONS> et </DO>, on peut retrouver les balises suivantes : <GO> ou <NOOP> ou <PREV> ou <REFRESH>

## **Balise <DO TYPE=DELETE>**

### 1. Description

L'instruction se trouvant entre les balises <DO TYPE=DELETE> et </DO> sera exécuté sur une opération de suppression de l'item courant d'une liste de choix.

### 2. Attributs

<b>type</b>	Renseigne une action prédéfinie.
Exemple	TYPE="DELETE"
<b>label</b>	Renseigne un label identifiant l'évènement. Ce label doit se restreindre à 6 caractères max (WML 1.1).
Exemple	LABEL="Delete"
<b>name</b>	Renseigne un nom à l'évènement.
Exemple	NAME="Supprime"
<b>optional</b>	Indique à l'utilisateur qu'il peut ignorer certains éléments.
Exemple	OPTIONAL="true   false"



## 3. Exemple

```
<DO TYPE=DELETE> LABEL="Delete">  
  <GO HREF="#deletecard"/>  
</DO>
```

## 4. Remarque

Entre la balise `<DO TYPE=DELETE>` et `</DO>`, on peut retrouver les balises suivantes : `<GO>` ou `<NOOP>` ou `<PREV>` ou `<REFRESH>`

## **Balise <DO TYPE=UNKNOWN>**

### 1. Description

L'instruction se trouvant entre les balises `<DO TYPE=UNKNOWN>` et `</DO>` sera exécuté comme un `<DO>` générique.

### 2. Attributs

<b>type</b>	Renseigne une action prédéfinie.
Exemple	TYPE="UNKNOWN"
<b>label</b>	Renseigne un label identifiant l'évènement. Ce label doit se restreindre à 6 caractères max (WML 1.1).
Exemple	LABEL="Go!"
<b>name</b>	Renseigne un nom à l'évènement.
Exemple	NAME="Lancer"
<b>optional</b>	Indique à l'utilisateur qu'il peut ignorer certains éléments.
Exemple	OPTIONAL="true   false"

### 3. Exemple

```
<DO TYPE=UNKNOWN> LABEL="Go!">  
  <GO HREF="#nextcard"/>  
</DO>
```

### 4. Remarque

Entre la balise `<DO TYPE=UNKNOWN>` et `</DO>`, on peut retrouver les balises suivantes : `<GO>` ou `<NOOP>` ou `<PREV>` ou `<REFRESH>`

## **Balise <EM>**

### 1. Description

Cette balise est utilisée pour formater du texte en caractère italique. Le texte se trouvant entre la balise `<EM>` et `</EM>` sera en caractère italique.

### 2. Attributs

Pas d'attributs.



## 3. Exemple

```
<EM>Mon premier site WAP</EM>
```

## 4. Remarque

Il est vivement conseillé d'utiliser cette balise plutôt que la balise `</>` car elle est dépendante du navigateur wap.

## **Balise <GO HREF="URL">**

### 1. Description

La balise `<GO>` définit une navigation entre URL. L'URL peut pointer sur une nouvelle card stockée sur un serveur, ou sur une card issue du même fichier. La balise `<GO>` apporte plus de puissance et de flexibilité pour la navigation.

### 2. Attributs

<b><i>accept-charset</i></b>	Renseigne une liste de caractères valides. Le serveur recevant la requête générée par <code>&lt;GO&gt;</code> doit accepter un des caractères d'encodage de la liste. Par défaut, sa valeur est <b>unknown</b> .
Exemple	<code>accept-charset="ISO-8859-1"</code>
<b><i>href</i></b>	Renseigne une URL à lancer.
Exemple	<code>HREF="/page.wml"</code>
<b><i>method</i></b>	Renseigne la méthode de soumission HTTP utilisée pour l'URL. GET fait apparaître les données lors de la transmission, contrairement à POST qui cache les données. Par défaut, la méthode est <b>GET</b> .
Exemple	<code>METHOD="POST"</code>
<b><i>sendreferer</i></b>	Si <code>sendreferer</code> est à <code>true</code> , l'utilisateur doit spécifier la passerelle wap en utilisant les entetes HTTP. Par défaut : <b>false</b> .
Exemple	<code>SENDREFERER="true   false"</code>

### 3. Exemple

```
<GO HREF="/page.wml" SENDREFERER="true"/>
```

### 4. Remarque

Entre la balise `<GO HREF="URL">` et `</GO>`, on peut retrouver les balises suivantes : `<POSTFIELD>`, `<SETVAR>`



## **Balise <HEAD>**

### **1. Description**

La balise <HEAD> contient des informations optionnelles sur le fichier WML. Ces informations optionnelles sont des contrôles d'accès et des informations meta.

### **2. Attributs**

Pas d'attributs.

### **3. Exemple**

```
<HEAD>  
  <ACCESS>, <META>  
</HEAD>
```

### **4. Remarque**

Entre la balise <HEAD> et </HEAD>, on peut retrouver les balises suivantes : <ACCESS>, <META>

## **Balise <I>**

### **1. Description**

Cette balise est utilisée pour formater du texte en caractère italique. Le texte se trouvant entre la balise </> et </> sera en caractère italique.

### **2. Attributs**

Pas d'attributs.

### **3. Exemple**

```
<I>Mon premier site WAP</I>
```

### **4. Remarque**

Pas de remarques particulières.



## **Balise <IMG SRC="URL">**

### 1. Description

Cette balise permet d'afficher une image au format WBMP sur l'écran de votre mobile. On peut spécifier la largeur, hauteur, alignement, etc.

### 2. Attributs

<b><i>Align</i></b>	Renseigne l'alignement de l'image par rapport au bord de l'écran (top middle  <b>bottom</b> ).
Exemple	ALIGN="MIDDLE"
<b><i>Alt</i></b>	Renseigne un nom à l'image. Ce nom pourrait être affiché à l'écran si le browser ne pouvait afficher l'image
Exemple	ALT="Ici une image"
<b><i>height</i></b>	Renseigne la hauteur de l'image en pixels ou en pourcentage par rapport à la taille de l'écran.
Exemple	HEIGHT="50%"
<b><i>hspace</i></b>	Indique le nombre de caractères blancs à gauche et à droite de l'image, exprimé en pixels ou en pourcentage.
Exemple	HSPACE="30%"
<b><i>localsrc</i></b>	Renseigne le nom d'une mémoire où serait mémoriser l'image. Cela permet de gagner en rapidité d'affichage.
Exemple	LOCALSRC="partcloudy"
<b><i>src</i></b>	Renseigne le chemin complet pour accéder à l'image.
Exemple	SRC="/images/pcloudy.wbmp"
<b><i>vspace</i></b>	Indique le nombre de caractères blancs en haut et en bas de l'image, exprimé en pixels ou en pourcentage.
Exemple	VSPACE="10"
<b><i>width</i></b>	Renseigne la largeur de l'image en pixels ou en pourcentage par rapport à la taille de l'écran.
Exemple	WIDTH="30"

### 3. Exemple

```
<IMG SRC="http://www.serveur.com/logowap.wbmp" ALT="Bienvenue sur serveur.com"/>
```

### 4. Remarque

Deux attributs sont obligatoires : *src* et *alt*. *src* est le chemin complet pour accéder à l'image. *alt* est le nom donné à l'affichage de l'image de telle sorte que si l'image ne peut s'afficher, le texte renseignant l'attribut alt soit affiché à la place.



## **Balise <INPUT TYPE=TEXT>**

### 1. Description

Cette balise est utilisée pour récupérer une saisie de l'utilisateur. Elle permet de rentrer des données qui sont sauvegardées dans une variable.

### 2. Attributs

<b><i>emptyok</i></b>	Indique si l'utilisateur peut laisser la zone de saisie vide (true). Par défaut : <b>false</b> , c'est à dire que l'utilisateur ne peut laisser la zone vide.
Exemple	EMPTYOK="true"
<b><i>format</i></b>	Renseigne un masque de saisie sur les caractères autorisés, par exemple : a pour des minuscules, A pour des majuscules.
Exemple	FORMAT="a"
<b><i>maxlength</i></b>	Renseigne le nombre de caractères possibles que l'utilisateur peut saisir sur son clavier.
Exemple	MAXLENGTH="15"
<b><i>name</i></b>	Renseigne un nom à la zone de saisie
Exemple	NAME="Nom"
<b><i>size</i></b>	Renseigne la taille en caractères de la zone de saisie.
Exemple	SIZE="10"
<b><i>tabindex</i></b>	Renseigne l'ordre d'affichage des zones de saisie, lors notamment de plusieurs zones de saisies.
Exemple	TABINDEX="1"
<b><i>title</i></b>	Renseigne un titre à la zone de saisie.
Exemple	TITLE="Saisie mot de passe"
<b><i>type</i></b>	Renseigne si la zone de saisie est normale ou cachée avec des étoiles, lors de la saisie d'un mot de passe par exemple (TEXT PASSWORD)
Exemple	TYPE="PASSWORD"
<b><i>value</i></b>	Renseigne une valeur par défaut qui sera affiché dans la zone de saisie.
Exemple	VALUE="5"

### 3. Exemple

```
<INPUT TYPE="TEXT" NAME="Nom" : " MAXLENGTH="15" EMPTYOK="TRUE" SIZE="10"/>
```

### 4. Remarque

En positionnant l'attribut *emptyok* à true, la zone de saisie est optionnelle pour l'utilisateur.

L'attribut format permet de réaliser un masque de saisie, idéal quand on veut uniquement saisir des chiffres ou des lettres !



## **Balise <INPUT TYPE=PASSWORD>**

### 1. Description

Cette balise est utilisée pour récupérer une saisie de l'utilisateur. Elle permet de rentrer des données qui sont sauvegardées dans une variable. La saisie sera cachée, c'est à dire que tous les caractères entrés au clavier seront remplacés à l'affichage (uniquement) par des caractères "\*". Idéal pour la saisie d'un mot de passe notamment !

### 2. Attributs

<b><i>emptyok</i></b>	Indique si l'utilisateur peut laisser la zone de saisie vide (true). Par défaut : <b>false</b> , c'est à dire que l'utilisateur ne peut laisser la zone vide.
Exemple	EMPTYOK="true"
<b><i>format</i></b>	Renseigne un masque de saisie sur les caractères autorisés, par exemple : a pour des minuscules, A pour des majuscules.
Exemple	FORMAT="a"
<b><i>maxlength</i></b>	Renseigne le nombre de caractères possibles que l'utilisateur peut saisir sur son clavier.
Exemple	MAXLENGTH="15"
<b><i>name</i></b>	Renseigne un nom à la zone de saisie
Exemple	NAME="Nom"
<b><i>size</i></b>	Renseigne la taille en caractères de la zone de saisie.
Exemple	SIZE="10"
<b><i>tabindex</i></b>	Renseigne l'ordre d'affichage des zones de saisie, lors notamment de plusieurs zones de saisies.
Exemple	TABINDEX="1"
<b><i>title</i></b>	Renseigne un titre à la zone de saisie.
Exemple	TITLE="Saisie mot de passe"
<b><i>type</i></b>	Renseigne si la zone de saisie est normale ou cachée avec des étoiles, lors de la saisie d'un mot de passe par exemple (TEXT PASSWORD)
Exemple	TYPE="PASSWORD"
<b><i>value</i></b>	Renseigne une valeur par défaut qui sera affiché dans la zone de saisie.
Exemple	VALUE="5"

### 3. Exemple

```
<INPUT TYPE="PASSWORD" NAME="Mot de passe" MAXLENGTH="15" EMPTYOK="FALSE" SIZE="10"/>
```

Mot de passe :

### 4. Remarque

En positionnant l'attribut *emptyok* à true, la zone de saisie est optionnelle pour l'utilisateur.

L'attribut format permet de réaliser un masque de saisie, idéal quand on veut uniquement saisir des chiffres ou des lettres !



## **Balise <META>**

### 1. Description

Cette balise permet, comme en HTML, de définir des méta informations génériques sur un fichier. On peut ainsi définir certaines informations, comme la version du fichier, son auteur, des mots clés et description pour les moteurs de recherche, etc.

### 2. Attributs

<b><i>http-equiv</i></b>	Définit un entete HTTP où le nom de l'entete et la valeur de l'attribut content sont définit dans l'entete.
Exemple	HTTP-EQUIV="STRING"
<b><i>name</i></b>	Renseigne le nom de la méta information.
Exemple	NAME="Author"
<b><i>forua</i></b>	Si cet attribut est positionné à true, la méta information sera délivré à l'utilisateur que si celui-ci supporte les balises META. Par défaut : false.
Exemple	FORUA="true false"
<b><i>content</i></b>	Renseigne le contenu de l'entete HTTP.
Exemple	CONTENT="Sebastien"
<b><i>scheme</i></b>	Renseigne une seconde information pour la balise META.
Exemple	SCHEME="Paul"

### 3. Exemple

```
<meta http-equiv="Content-Type" content="text/vnd.wap.wml">  
<meta name="Keywords" content="wap,wml,developpement">
```

### 4. Remarque

Ces balises se placent avant la *balise <WML>*. Elles sont très utilisées par les moteurs de recherche.

## **Balise <NOOP>**

### 1. Description

Cette balise définit une opération qui ne fait rien suivre à un évènement.

### 2. Attributs

Pas d'attributs.

### 3. Exemple

```
<NOOP/>
```

### 4. Remarque

Pas de remarques particulières.



## **Balise <ONEVENT TYPE=ONEVENTFORWARD>**

### **1. Description**

Cette balise définit un évènement qui sera appelé lorsque l'utilisateur provient d'une card depuis une balise <GO> ou autre mécanisme de navigation.

### **2. Attributs**

Pas d'attributs.

### **3. Exemple**

```
<ONEVENT TYPE=ONEVENTFORWARD>  
  <REFRESH>  
    <SETVAR NAME="var1" value="" />  
    <SETVAR NAME="var2" value="" />  
  </REFRESH>  
</ONEVENT>
```

Ici, si ce bout de code était à l'intérieur d'une card (<CARD> et </CARD>), à chaque entrée dans la card, les variables var1 et var2 seraient remises à une valeur vide (null). Par contre, les variables ne sont pas modifiées si la card est appelée depuis une tâche de retour en arrière (<PREV>) par exemple...

### **4. Remarque**

Entre la balise <ONEVENT> et </ONEVENT>, on peut trouver les balises suivantes : <GO>, <NOOP>, <PREV> et <REFRESH>.

## **Balise <ONEVENT TYPE=ONEVENTBACKWARD>**

### **1. Description**

Cette balise définit un évènement qui sera appelé lorsque l'utilisateur provient d'une card depuis une balise <GO> ou autre mécanisme de navigation.

### **2. Attributs**

Pas d'attributs.

### **3. Exemple**

```
<ONEVENT TYPE=ONEVENTBACKWARD>  
  <REFRESH>  
    <SETVAR NAME="var1" value="" />  
    <SETVAR NAME="var2" value="" />  
  </REFRESH>  
</ONEVENT>
```

Ici, si ce bout de code était à l'intérieur d'une card (<CARD> et </CARD>), à chaque entrée dans la card provenant d'un retour en arrière dans la navigation de l'utilisateur, les variables var1 et var2 seraient remises à une valeur vide (null). Par contre, les variables ne sont pas modifiées si la card est appelée pour la première fois et n'est donc pas dans l'historique du cache du portable.



## 4. Remarque

Entre la balise `<ONEVENT>` et `</ONEVENT>`, on peut trouver les balises suivantes : `<GO>`, `<NOOP>`, `<PREV>` et `<REFRESH>`.

### **Balise `<ONEVENT TYPE=ONTIMER>`**

#### 1. Description

Cette balise définit un évènement qui sera appelé lorsque le temps est expiré et ce depuis une balise `<TIMER>`.

#### 2. Attributs

Pas d'attributs.

#### 3. Exemple

```
<ONEVENT TYPE=ONTIMER>
  <GO HREF="#card2"/>
</ONEVENT>
<TIMER VALUE="20"/>
```

Ici, si ce bout de code était à l'intérieur d'une card (`<CARD>` et `</CARD>`), lorsque le temps exprimé dans la balise `TIMER` expire, l'évènement est activé et exécuté.

## 4. Remarque

Entre la balise `<ONEVENT>` et `</ONEVENT>`, on peut trouver les balises suivantes : `<GO>`, `<NOOP>`, `<PREV>` et `<REFRESH>`.

### **Balise `<ONEVENT TYPE=ONPICK>`**

#### 1. Description

Cette balise définit un évènement qui sera appelé lorsque l'utilisateur aura sélectionné un item dans un menu de choix `<OPTION>`.

#### 2. Attributs

Pas d'attributs.

#### 3. Exemple

```
<ONEVENT TYPE=ONPICK>
  <GO HREF="#card2"/>
</ONEVENT>
<SELECT NAME="Sport">
  <OPTION VALUE="Football"> Football </OPTION>
  <OPTION VALUE="Tennis"> Tennis </OPTION>
</SELECT>
```

Ici, lorsque l'utilisateur sélectionnera un item dans le choix qui lui est proposé, un évènement `ONPICK` sera déclenché en interprété.



## 4. Remarque

Entre la balise `<ONEVENT>` et `</ONEVENT>`, on peut trouver les balises suivantes : `<GO>`, `<NOOP>`, `<PREV>` et `<REFRESH>`.

## **Balise `<OPTGROUP>`**

### 1. Description

Permet de réaliser une hiérarchie entre les éléments `<OPTION>`. Ainsi les éléments `<OPTION>` peuvent être regroupés sous une même entité.

### 2. Attributs

<b>title</b>	Renseigne un titre à l'entité qui regroupe toutes les OPTION.
Exemple	TITLE="Sport"

### 3. Exemple

```
<OPTGROUP Title="Sports">
  <OPTION ONPICK="ski.wml" VALUE="ski">Ski</OPTION>
  <OPTION ONPICK="tennis.wml" VALUE="tennis">Tennis</OPTION>
  <OPTION ONPICK="football.wml" VALUE="football">Football</OPTION>
</OPTGROUP>
```

## 4. Remarque

On peut retrouver entre la balise `<OPTGROUP>` et `</OPTGROUP>` les balises suivantes : `<OPTION>`, `<OPTGROUP>`.

## **Balise `<OPTION>`**

### 1. Description

Cette balise s'utilise pour proposer un choix dans une sélection. L'utilisateur peut donc faire son choix.

### 2. Attributs

<b>Onpick</b>	Renseigne une URL qui sera lancée si l'utilisateur sélectionne cet item dans la sélection.
Exemple	ONPICK="http://serveur.com"
<b>Title</b>	Renseigne un titre au choix et sera ainsi affiché (très peu utilisé).
Exemple	TITLE="choix1"
<b>Value</b>	Renseigne une valeur à la variable.
Exemple	VALUE="1"

### 3. Exemple

```
<SELECT NAME="Jour">
  <OPTION ONPICK="lundi.wml" VALUE="Lundi">Lundi</OPTION>
  <OPTION ONPICK="mardi.wml" VALUE="Mardi">Mardi</OPTION>
  <OPTION ONPICK="mercredi.wml" VALUE="Mercredi">Mercredi</OPTION>
</SELECT>
```



## 4. Remarque

On peut retrouver entre la balise `<OPTION>` et `</OPTION>` les balises suivantes : `<ONEVENT>` et du texte.

## **Balise <P>**

### 1. Description

Cette balise permet de définir un paragraphe avec un alignement à paramétrer. Cette balise est très utilisée en WML.

### 2. Attributs

<b>Align</b>	Renseigne l'alignement du texte qui suivra la balise ( <b>left</b>  center right).
Exemple	ALIGN="CENTER"
<b>Mode</b>	Prend comme valeur <b>wrap</b> ou nowrap. Si l'attribut mode est positionné à nowrap, le texte affiché débordera de l'écran si celui-ci est trop grand. Par défaut : wrap, c'est à dire que le texte est remis à la ligne automatiquement.
Exemple	MODE="WRAP"

### 3. Exemple

```
<P ALIGN="CENTER" MODE="WRAP" >  
</P>
```

## 4. Remarque

La balise `<P>` s'utilise généralement après une balise `<CARD>` pour débiter une nouvelle card.

## **Balise <POSTFIELD>**

### 1. Description

Cette balise permet de passer des variables avec une valeur associée lorsque le serveur reçoit une requête `<GO>`.

### 2. Attributs

<b>Name</b>	Renseigne le nom de la variable.
Exemple	NAME="var1"
<b>Value</b>	Renseigne une valeur à la variable.
Exemple	VALUE="1"

### 3. Exemple

```
<POSTFIELD NAME="jour" VALUE="lundi"/>
```

## 4. Remarque

Les balises `<POSTFIELD>` s'utilisent avec la balise `<GO>`.



## **Balise <PREV>**

### **1. Description**

Cette balise répond à un évènement. Le programme se trouvant entre la balise <PREV> et </PREV> sera exécuté lorsque l'évènement associé à <PREV> sera lancé. Cet évènement intervient lorsque le navigateur retourne à un écran précédent.

### **2. Attributs**

Pas d'attributs.

### **3. Exemple**

```
<PREV>  
  <SETVAR>  
</PREV>
```

### **4. Remarque**

Entre la balise <PREV> et </PREV>, on peut trouver la balise suivante : <SETVAR>.

## **Balise <REFRESH>**

### **1. Description**

Cette balise permet d'actualiser le contenu avant l'affichage de la card.

### **2. Attributs**

Pas d'attributs.

### **3. Exemple**

```
<REFRESH>  
  <SETVAR NAME="var1" VALUE="val1"/>  
</REFRESH>
```

### **4. Remarque**

Cette balise s'utilise systématiquement avec la balise <SETVAR>.



## **Balise <SELECT>**

### 1. Description

Cette balise permet de créer une liste de choix qui sera proposé à l'utilisateur. Celui-ci pourra ainsi sélectionner tel ou tel élément de cette liste.

### 2. Attributs

<b><i>iname</i></b>	Renseigne le nom de la variable qui va mettre à jour l'index de sélection.
Exemple	INAME="val1"
<b><i>ivalue</i></b>	Renseigne l'index par défaut à sélectionner dans la liste de choix. Cet attribut n'est traité si l'attribut INAME n'est pas affecté.
Exemple	IVALUE="1"
<b><i>multiple</i></b>	Renseigne si l'utilisateur peut réaliser plusieurs choix dans la liste (true) ou un unique (false). Par défaut false.
Exemple	MULTIPLE="true false"
<b><i>name</i></b>	Renseigne le nom de la variable qui sera affecté de la sélection de l'utilisateur.
Exemple	NAME="var2"
<b><i>tabindex</i></b>	Renseigne une tabulation relative.
Exemple	TABINDEX="2"
<b><i>title</i></b>	Renseigne un titre pour la sélection proposée à l'utilisateur et qui sera affichée.
Exemple	TITLE="Votre choix"
<b><i>value</i></b>	Renseigne par défaut une valeur à la variable définie dans l'attribut NAME.
Exemple	VALUE="5"

### 3. Exemple

```
<SELECT NAME="hobbies" MULTIPLE="true">  
  <OPTION VALUE="ski">Ski</OPTION>  
  <OPTION VALUE="lire">Lire</OPTION>  
  <OPTION VALUE="film">Regarder un film</OPTION>  
</SELECT>
```

### 4. Remarque

Entre la balise <SELECT> et </SELECT>, on retrouve les balises suivantes : <OPTION>, <OPTGROUP>.

La balise *SELECT* s'utilise systématiquement avec la balise *OPTION*.



## **Balise <SETVAR>**

### 1. Description

Cette balise permet de définir une variable avec une valeur associée. Cela est utile lorsque l'utilisateur passe de card en card, on peut ainsi transporter des variables.

### 2. Attributs

<b>name</b>	Renseigne un nom pour la variable.
Exemple	NAME="var1"
<b>value</b>	Renseigne une valeur à la variable.
Exemple	VALUE="5"

### 3. Exemple

```
<GO HREF="#card2">  
  <SETVAR NAME="variable1" VALUE="5"/>  
</GO>
```

### 4. Remarque

Cette balise s'utilise avec les balises suivantes : <PREV>, <REFRESH>, <GO>.

## **Balise <SMALL>**

### 1. Description

Cette balise est utilisée pour formater du texte en caractère petit. Le texte se trouvant entre la balise <SMALL> et </SMALL> sera en petit caractère.

### 2. Attributs

Pas d'attributs.

### 3. Exemple

```
<SMALL>Mon premier site WAP</SMALL>
```

## **Balise <STRONG>**

### 1. Description

Cette balise est utilisée pour formater du texte en caractère gros. Le texte se trouvant entre la balise <STRONG> et </STRONG> sera en caractère large.

### 2. Attributs

Pas d'attributs.

### 3. Exemple

```
<STRONG>Portail dumobile.com</STRONG>
```

### 4. Remarque

Il est vivement conseillé d'utiliser cette balise plutôt que la balise <BIG> car elle est dépendante du navigateur wap.



## **Balise <TABLE>**

### 1. Description

Cette balise permet de définir une table où l'on indique le nombre de colonnes, le titre et l'alignement.

Une table permet de structurer l'affichage des informations.

### 2. Attributs

<b><i>align</i></b>	Renseigne l'alignement de la table par rapport à l'écran ("L" pour left, "C" pour center et "R" pour right).
Exemple	ALIGN="L"
<b><i>columns</i></b>	Renseigne le nombre de colonnes dans la table.
Exemple	COLUMNS="2"
<b><i>title</i></b>	Renseigne un titre à la table.
Exemple	TITLE="Ma table"

### 3. Exemple

```
<TABLE COLUMNS="1">  
  <TR>  
    <TD>Un element...</TD>  
  </TR>  
</TABLE>
```

## **Balise <TD>**

### 1. Description

Cette balise définit une nouvelle colonne dans une table. On trouve la balise <TD> dans une balise <TR>.

### 2. Attributs

Pas d'attributs.

### 3. Exemple

```
<TABLE COLUMNS="2">  
  <TR>  
    <TD>Un element...</TD>  
    <TD>Un autre element...</TD>  
  </TR>  
</TABLE>
```

### 4. Remarque

Entre la balise <TD> et </TD>, on peut retrouver les balises suivantes : <A>, <ANCHOR>, <BR>, <IMG>, <TABLE>, et du texte.



## **Balise <TEMPLATE>**

### 1. Description

Cette balise permet de définir des actions qui seront appliquées à toutes les cards contenues dans le fichier WML. Typiquement, la balise <TEMPLATE> est utilisée pour définir des actions qui se répètent tout au long de la navigation entre les cards.

### 2. Attributs

<b>oneventbackward</b>	Renseigne une url qui sera lancée à chaque retour sur la card précédente.
Exemple	ONEVENTBACKWARD="URL"
<b>oneventforward</b>	Renseigne une url qui sera lancée à chaque sortie d'une card.
Exemple	ONEVENTFORWARD="URL"
<b>ontimer</b>	Renseigne une URL qui sera lancée après une durée définie dans une balise <EVENT>.
Exemple	ONTIMER="URL"

### 3. Exemple

```
<TEMPLATE>  
<DO TYPE="options" LABEL="Help">  
  <GO HREF="#helpcard"/>  
</DO>  
</TEMPLATE>
```

### 4. Remarque

Entre la balise <TEMPLATE> et </TEMPLATE>, on peut retrouver les balises suivantes : <DO>, <ONEVENT>.

## **Balise <TIMER>**

### 1. Description

Cette balise permet de définir une temporisation dans l'exécution. Cette balise s'utilise étroitement avec la balise <ONEVENT TYPE="ontimer">. Au bout d'une certaine valeur définie à travers l'attribut VALUE, une card est exécutée. Une card peut seulement avoir un timer et vice-versa.

### 2. Attributs

<b>name</b>	Renseigne un nom optionnel au timer.
Exemple	NAME="timer1"
<b>value</b>	Renseigne une valeur en seconde. Cette valeur est utilisée si l'attribut NAME est vide ou non présent, sinon elle est ignorée.
Exemple	VALUE="20"



### 3. Exemple

```
<ONEVENT TYPE="ontimer">  
  <GO HREF="#card2"/>  
</ONEVENT>  
<TIMER VALUE="20"/>
```

### 4. Remarque

Cette balise s'utilise avec la balise `<ONEVENT TYPE="ontimer">`.

## **Balise <TR>**

### 1. Description

Cette balise définit une nouvelle ligne dans une table. On trouve la balise `<TR>` dans une balise `<TABLE>`.

### 2. Attributs

Pas d'attributs.

### 3. Exemple

```
<TABLE>  
  <TR>  
    <TD>Un element...</TD>  
  </TR>  
</TABLE>
```

### 4. Remarque

Entre la balise `<TR>` et `</TR>`, on retrouve une ou plusieurs balises suivantes : `<TD>`.

## **Balise <U>**

### 1. Description

Cette balise est utilisée pour formater du texte en caractère souligné. Le texte se trouvant entre la balise `<U>` et `</U>` sera en caractère souligné.

### 2. Attributs

Pas d'attributs.

### 3. Exemple

```
<U>Mon premier site WAP</U>
```

### 4. Remarque

Pas de remarques particulières.



## **Balise <WML>**

### **1. Description**

Cette balise est utilisée dans tous les fichiers WML. La balise ouvrante <WML> est définie en début de fichier et la balise fermante </WML> en fin de fichier. Cette balise peut contenir une ou plusieurs balises <CARD>.

### **2. Attributs**

Pas d'attributs.

### **3. Exemple**

```
<WML>
<CARD>
  <P>
    Ceci est une card.
  </P>
</CARD>
</WML>
```

### **4. Remarque**

Cette balise est utilisée une fois et pour chaque fichier WML.

## **14.0 Bibliographie**

Ce document est une compilation de textes trouvés sur le Web.