

WAP (wml) 开发教程

WAP (无线应用协议) 是在数字移动电话、因特网或其他个人数字助理机 (PDA)、计算机应用之间进行通讯的开放全球标准。它是由一系列协议组成, 用来标准化无线通信设备, 可用于 Internet 访问, 包括收发电子邮件, 访问 WAP 网站上的页面等等。

WAP 将移动网络和 Internet 以及公司的局域网紧密地联系起来. 提供一种与网络类型、运行商和终端设备都独立的移动增值业务。

通过这种技术, 无论你在何地、何时只要你需要信息, 你就可以打开你的 WAP 手机, 享受无穷无尽的网上信息或者网上资源。如: 综合新闻、天气预报、股市动态、商业报道、当前汇率等。电子商务、网上银行也将逐一实现。你还可以随时随地获得体育比赛结果、娱乐圈趣闻以及幽默故事, 为生活增添情趣, 也可以利用网上预定功能, 把生活安排的有条不紊。

WAP 协议包括以下几层:

- 1、Wireless Application Environment (WAE)
- 2、Wireless Session Layer (WSL)
- 3、Wireless Transaction Layer (WTP)
- 4、Wireless Transport Layer Security (WTLS)
- 5、Wireless Transport Layer (WDP)

其中, WAE 层含有微型浏览器、WML、WMLSCRIPT 的解释器等等功能。WTLS 层为无线电子商务及无线加密传输数据时提供安全方面的基本功能。

WAP 协议的诞生是 WAP 论坛成员多年努力的结果。它是针对不同的协议层定义了一系列协议, 这些协议使得各方面的厂商和公司可以协同工作, 开发无线通信网络的应用。目前有超过 100 个成员加入 WAP 论坛, 包括有终端和基础设备的制造商, 有移动通信的网络运营商, 有业务提供商, 有软件公司, 也有网络内容提供商等, 共同为移动设备开发服务和应用。

但是, 目前由于无线网的带宽等等因素的限制, WAP 手机的在多媒体上的应用如: 可视会议、多媒体教学等等, 还须一段时日。

WAP 规范还在不断地完善, WAP 论坛成员们在加紧开发功能完善的 WAP 设备, 这无疑加快其在无线因特网综合服务领域的扩展速度。

关于 WAP 有几点, 应该注意一下:

1.WAP 不只是将手机带入 IP/Internet 网络

WAP 规范考虑了很多无线网络和手机的特性。即便是 3G 的手机也无法把屏幕做的象通用计算机一样, 来浏览 HTML 页面。直接接入 IP 网, 9110 在 98 年就可以, 但是应用呢, 不可能 Nokia, Motorola, Ericsson 推出手机时各自定义一套应用。

象 WWW 一样, 在规范推出以前, 已经有了很多基于 IP 网络的应用, 包括一些做的很好的软件。但 Internet 遍地开花却是在 WWW 规范推出之后。WAP 很大的意义在于统一的标准。

WAP 和无线 IP 网络并不矛盾, WAP 协议规定, 若承载基于 IP, WDP 就是 UDP。等无线 IP 网络普及, 若认为上面几层太繁琐, 最多是 WAP 协议中间几层略掉(其实也没太多东西)。

2.WAP 应用模式很多

只要手机和网络支持统一标准, 业务发展是不可限量的。这一点就不多说了。

3.WAP 是发展的

WAP 规范不仅定义了和 Internet 结合部分，还定义 WTE，及语音部分，可以说发展还早呢。要是以后通讯核心网建起来，语音业务没准都采用 WAP 来实现。

WML 学习 (一)

概述

WML (Wireless Markup Language - 无线标记语言) 这种描述语言同我们常听说的 HTML 语言同出一家，都属于 XML 语言这一大家族。HTML 语言写出的内容，可以在我们的 PC 机上用 IE 或是 Netscape 等浏览器进行阅读，而 WML 语言写出的文件则是专门用来在手机等的一些无线终端显示屏上显示，供人们阅读的，并且同样也可以向使用者提供人机交互界面，接受使用者输入的查询等信息，然后向使用者返回他所想要获得的最终信息。

基本规则

* WML 使用 XML 文档字符集，目前支持 Unicode 2.0，和 HDML 不同，WML 的所有标签，属性和规定的可接收值必须小写，CARD 的名字和变量也是区分大小写的。和 HDML 一样，对于连续的空字符，只显示一个空格。标签内属性的值必须用"或者'括起来，属性名，= 和值之间不能有空格。对于不成对出现的标签，必须在 > 前加 /，比如
。在对变量的引用上和 HDML 基本相同，有：

`$(var1:esc)`

`$(var1:noesc)`

`$(var1:unesec)`

三种形式，具体含义参见前面 HDML 的相关部分。在对保留字符的处理上也基本相同，对应的取代字符有：

`<<`

`>>`

`' '`

`" "`

`& &`

`$ $$`

空格

`--`

这里要指出的是在 URL 的传递过程中，用来连接参数的 & 必须转化为 & 。

WML 学习 (二)

基本格式和文件头

wml 文件的一般格式：

```
<?xml version="1.0"?>
```

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
```

```
  <head>
```

```
    <access/>
```

```
    <meta.../>
```

```

</head>
<card>
  Some contents...
</card>
<wml>

```

结构看上去和 HTML 文件很类似，对于每一个 DECK，在其文档开头必须指明以下的类型声明：

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">

```

千万注意字母的大小写。对于一个 DECK，其文件大小最好不要超过 1.2K。

<wml>标签和 HTML 中的<html>标签一样，用来表明这是一个 WML 的 DECK，它有一个可选的 xml:lang 属性来制定文档的语言，比如<wml xml:lang="zh">表示文档语言为中文。

和 HTML 一样<head>标签包含了该 DESK 的相关信息。<head> 标签之间可以包含一个<access>标签和多个<meta>标签。

<access domain="域" path="/路径" />相当于 HTML 中的<BASE>标签，指定该 DECK 的访问控制信息，它用两个可选的属性，domain 用来指定域，默认值为当前域， path 用来指定路径，默认值为"/"，即根目录。由于 <access>单独使用，所以要用/结尾，以后对于类似的情况不再赘述。

<meta 属性 content="值" scheme="格式" forua="true|false"/> 和 HTML 中的类似，提供了该 DECK 的 meta 信息。属性是必选的，包括一下三种情况 name="name" UP.Link Server 忽略 meta 数据 http-equiv="name" UP.Link Server 将 meta 数据转为 HTTP 响应头（同 HTML）

user-agent="agent" UP.Link Server 直接将 meta 数据传给手机设备， content 属性也是必选的，其内容根据属性而定。scheme 属性目前尚不支持。forua 为可选属性，指定在该 wml 文件传到客户端之前，<meta>标签是不是被中间代理删除（因为传输的协议可能改变），默认值为 false。

目前支持的 meta 数据：

<meta http-equiv="Cache-Control" content="max-age=3600"/>指定 DECK 在手机内存缓存中的存储时间段，默认的为 30 天(除非内存耗尽)，在该期间，手机对于访问过的 DECK 直接从缓存里调用。如果信息是对时间敏感的，可以用 max-age 指定 DECK 在缓存里的生存期，最小单位是秒，如果指定为 0，则每次都需通过连接服务器来调用该 DECK。

<meta user-agent="vnd.up.markable" content="false"/>和<meta user-agent="vnd.up.bookmark" content="指定的 URL"/> 类似于普通浏览器的书签功能。当用户将一个 CARD 做了书签后，手机浏览器首先用一个标记记录该 CARD，这个标记默认的是<card>标签中的 title 属性(以后会讲到)，然后当用户选择了该书签以后，浏览器就会打开被记录的 URL。但是因为默认的情况下，手机会记录所有的 DECK，所以，一般<meta>被用来使手机不要记录当前的 URL，即<meta user-agent="vnd.up.markable" content="false"/>。此外，如果要为书签指定不同于当前 DRECK 的 URL，用<meta user-agent="vnd.up.bookmark" content="指定的 URL"/>。

一个 DECK 可以包含多个 CARD，每个 CARD 的内容可能不止一屏显示，注意 DECK，CARD 和屏幕显示范围的关系。一个 CARD 用<card>和</card>包含。<card>可以包含以下可选的属性：

```

<card id="name" title="label" newcontext="false" ordered="true" onenterforward="url" onenterbackward="url"
ontimer="url">

```

*id 属性用来指定 CARD 的名字，可用来在 CARD 间跳转，相当于在 HTML 中在页内跳转时用。

*title 属性用来作为书签的标记，该属性一般不会显示在屏幕上。

*newcontext 属性，默认值为 false，用来指示当跳转到本 CARD 时，手机是不是要清除以前保留的信息，包括变量，堆栈里的历史记录，重新设置手机状态等。

*ordered 属性，默认值是 true，表明该 CARD 里的内容是按固定的顺序显示，还是按用户的选择来显示。这点和 HTML 不同，CARD 页面里的内容可以按一定的顺序显示，默认的是按线性顺序显示，即按代码的顺序，但是，要注意的是，以下三个标签必须按以下顺序来写<onevent><timer><do>，（这和以后要讲的“事件”有关）。这样做是为了方便填表单，当 ordered 设置为 true 时，如果一个表单的内容不能在一屏里显示完，就分成多屏显示；当 ordered 设置为 false 时，手机可以显示一个概要 CARD 来总结有效的选项，用户可以从中选取表单选项来填写。

*onXXX 属性，类似于 HTML 标签中的 onXXX 属性，用来捕捉事件，当事件被触发时执行指定的操作(任务)，在这里是转向某个 URL。

WML 学习 (三)

显示文本

在文本的显示上 WML 基本和 HTML 相同。文字段落包含在<p align="alignment" mode="wrapmode">和</p>之间，align 属性指定该段文字的对齐方式，默认的是 left，其他可选择 right 和 center；mode 属性指定当一行显示不下所有的文字时是否自动换行，默认的是自动换行 wrap，如果选 nowrap，则在一行中显示，浏览器会通过类似于水平滚动条的机制来显示所有文字。

换行标签也一样为
，这里先替一下，在标单中如果有多个<input>或者<select>，其间不要用
，否则会使手机浏览器认为是断点而把表单分页显示。

文字的修饰标签有、<i>、<u>、、、<big>和<small>，意义和 HTML 里的相同。

表格的显示标签也和 HTML 相近，使用<table title="name" align="left|right|center" columns="列数">、<tr>和<td>来显示，<table>的 title 属性用来给表格取个名字，columns 属性指定表的列数，不能为 0，可选的 align 属性和前面提到的一样是对齐方式。表格中可以包含文字和图片。

test1.wml

```
-----
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card>
<p align="center">
<i>Hello</i><br/>
<b><i>World!</i></b>
<table title="mytable" align="right" columns="2">
<tr>
<td>1-1</td>
<td>1-2</td>
</tr>
```

```

<tr>
<td>2-1</td>
<td>2-2</td>
</tr>
</table>
</p>
</card>
</wml>

```

显示图片

显示图片(1 位黑白 BMP 图片)的标签一样类似于 HTML, ``, 属性中 `alt` 和 `src` 是必须要有的, 其他可选。另外要注意的是 `` 要放在 `<p>` 里, 不能放在 `<do>` 和 `<option>` 等功能键标签和选单标签里。

*`alt` 属性用来指定当手机不支持图片显示时用来显示的文字。

*`src` 属性指定图片的 URL, 但当有了以下的 `localsrc` 属性时, 手机浏览器就忽略 `src` 属性。

*`localsrc` 属性用来指定显示存在手机 ROM 中的图标, 如果找不到, 则到 UP.Link Server 上去找。

*可选的 `align` 属性用来表明图片同当前行文本的对齐方式, 默认值为 `bottom`, 可选 `to` 和 `middle`。

*`height`、`width`、`vspace`、`hspace` 属性分别指定图片的长宽和距环绕文字的间距, 目前不是所有的 WAP 手机都支持这些属性。

test2.wml

```

-----
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org
/DTD/wml_1.1.xml">
<wml>
<card>
<p align="center"></p>
</card>
</wml>

```

WML 学习 (四)

锚和任务

连接是 HTML 页面里最基本的功能, 在 WML 里也一样用 `` 和 `` 来包括用来建立连接的文字, 必选属性 `href` 指定了要打开的 URL, 可选的 `title` 属性给该连接取个标记名字, 这个名字将作为软按钮之一的 ACCEPT 键(详见以前的 HDML 入门文章)的标记显示在屏幕的软按钮区, 所以通常可以将属性作为提示文字是用。

然而, 以上的连接在 WML 里只是任务的一种情况, 为了能够使用其它任务, 所以引进了新的标签 `<anchor title="label">任务标签 文本</anchor>`, `<a>` 其实是当任务标签为 `<go/>` 时的简单表示方式。

test3.wml

```

-----
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org
/DTD/wml_1.1.xml">
<wml>
<card>
<p>
<anchor title="Link1"><go href="test1.wml"/>News</anchor><br/>
<a title="Link2" href="test2.wml">Sports</a>
</p>
</card>
</wml>

```

WML 里的任务标签有以下几种，除了用于<anchor>，还可以用在事件中：

1)<go>用来指示浏览器显示指定的 URL，包括 DECK 的第一个 CARD 或者指定的 CARD。语法如下：

```

<go href="url" sendreferer="false|true" method="get|post" accept-charset
="charset">
  <postfield name="name" value="value"/>
  <setvar name="name" value="value"/>
</go>

```

其中 href 属性为必选，其他为可选。sendreferer 属性表示是否传递调用 href 所指定的 URL 的页面的 URL，也就是当前页的 URL，即 HTTP 头中的 HTTP_REFERER，默认值为 false，可选值为 true。method 和 HTML 中的表单 FORM 的 method 属性一样，指定表单是以 get 的方式还是 post 的方式递交，以便 cgi 处理，默认值为 get，但如果未指定 method 但<go></go>间存在<postfield>，手机自动以 post 方式传递。accept-charset 属性可覆盖在 HTTP 头里指定的字符集，可以写多个字符集，如 accept-charset="UTF-8,US-ASCII,ISO-8859-1"。

<postfield name="name" value="value"/>可以看作是 HTML 表单 FORM 中的<INPUT TYPE="HIDDEN" NAME="变量名" VALUE="值">，通过它可以向指定的 URL 传递以“变量名/值”形式的数据。name 和 value 属性都是必选的。注意只有这里的变量是用来递交给 CGI 程序的。

除了<postfield>，还可以在<go>和</go>间加入一句或者多句<setvar name="name" value="value"/>，该语句的意思是，当触发某一事件时，给变量赋值。

要注意的是，当<go>和</go>之间没有任何语句时，要用<go/>的形式，这点比较特别，比如<anchor title="Link1"><go href="test.wml"/>News</anchor>。

2)<prev>用来将当前页面的 URL 压入 URL 历史堆栈，并打开此前的 URL，若该 URL 不存在，则<prev>无效。语法类似<go>：<prev><setvar name="name" value="value"/></prev>，<prev>和</prev>之间可加入一句或多句<setvar name="name" value="value"/>，若不加，则必须变成<prev/>的形式。

3)<refresh>用来刷新当前的页面，从而使得页面内的变量刷新，语法为<refresh>

<setvar name="name" value="value"/></refresh>。

4) <noop>, 表示什么也不做, 该标签不能用在<anchor>中, 一般用在覆盖 DECK 级的 <do>(以后会说明)。

WML 学习 (五)

显示表单

类似于 HTML 的<FORM>, <fieldset>可用来包括一组表单选项, 但不是必须的。前面说过, 当<card>的 ordered 设置为 false 时, 手机可以显示一个概要 CARD 来总结有效的选项, 方便用户从中选取表单选项来填写, 概要 CARD 就是根据<fieldset>和独立的输入框<input>以及选单<select>来总结的。语法为, <fieldset title="label">表单内容</fieldset>, 可选的 title 属性除了可用来表示表单的名字外, 还在概要 CARD 里作为选择项的标题。表单内容可以是嵌套的<fieldset>, 输入框<input>, 选单<select>和必要的提示文本。 <input name="variable" title="label" type="type" value="value" default="default" format="specifier" emptyok="false|true" size="n" maxlength="n" tabindex="n"/>, 用来输入文本, 除了 name 属性是必要的, 其他可选。

*name 属性, 指定了用来存储该输入文本的变量名字。

*title 属性, 该输入框的名字, 同样可被用来作为概要 CARD 页中的选项名。

type 属性, 默认值为 text, 如选择 password, 则输入的数据显示为。

*value 属性, 在语法和行为上等同于下面的 default 属性,

*default 属性, 指定输入框的默认值, 即 name 属性指定变量的默认值, 当用户输入新值时, 该值无效, 如果该值不符合以下 format 属性的规定, 则手机也忽略该默认值。

*format 属性, 用来格式化输入的数据, 可用的标记如下, 使用时可用“一位数字标记”和“*标记”的形式, 前者代表 N 个标记型字符, 如 3X, 后者代表任意个(小于 maxlength 属性的值)标记型字符。

标记	描述
A	任何符号或者大写字母(不包括数字)
a	任何符号或者小写字母(不包括数字)
N	任何数字(不包括符号或者字母)
X	任何符号、数字或者大写字母(不可改变为小写字母)
x	任何符号、数字或者小写字母(不可改变为大写字母)
M	任何符号、数字或者大写字母(可改变为小写字母)或者多个字符, 默认为首字大写
m	任何符号、数字或者小写字母(可改变为大写字母)或者多个字符, 默认为首字小写

*maxlength 属性, 指定了用户可输入的最大字符长度, 最大限制为 256 个字符。

*emptyok 属性, 表示用户可否不填输入框, 默认为 false, 即要填。

*size 属性, 输入框显示长度, 目前未被支持。

*tabindex 属性, 类似于在 HTML 表单中按 TAB 键后, 焦点落在哪个选项上, 该值决定了这个选择顺序, 数字大的排在后面。目前未被支持。

test4.wml

```
<?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="card1" ordered="false">
<p>
USERNAME:<input name="userName" title="User Name" type="text" value="YourNameHere" format="*M" emptyok="false" maxlength="12" tabindex="1"/>
PASSWORD:<input name="password" title="Password" type="password" format="8x" emptyok="false" maxlength="8" tabindex="2"/>
COMMENTS:<input name="comments" title="comment" type="text" value="YourCommentsHere" format="*M" emptyok="true" maxlength="30" tabindex="3"/>
</p>
</card>
</wml>

```

将上例<card>中的 ordered 的值改为 "true" 试试看, 然后再在<p>和</p>内加入<fieldset title="field1">和</fieldset>试试看。

<select>选单类似于 HTML 表单中的<SELECT>, <select>和</select>间可包含<optgroup>和<option>标签, 语法如下, 所有属性都为可选:

```

<select title="label" multiple="false|true" name="variable" default="default" iname="index_var" ivalue="default"
tabindex=" n">
<optgroup title="label">选单内容</optgroup>
<option title="label" value="value" onpick="url">
事件或者文本
</option>
</select>

```

*title 属性, 如以上<input>的 title 属性。

*multiple 属性, 指定用户可否进行多项选择, 默认值为 false。

*name 属性, 用来存储用户选择项的变量名, 其值为 <option>标签的 value 属性, 若用户没有选择又没有用 default 属性指定默认值, 则手机将改变量赋值为空字符串 "", 对于多项选择, 每个值用“;”分开。

*default 属性, 可为 name 属性指定的变量赋默认值。

*iname 属性, 用来记录用户选择项的位置, 相应的值从 1 开始。若没选, 则该值为 0。

*ivalue 属性, 用来记录默认值所在的位置。

<optgroup>可用来将多个<option>分组, <optgroup>和</optgroup>间还可包括<optgroup>和<option>, 该标签目前尚未被支持。

<option>, 类似于 HTML 中选单的<OPTION>, 用来表示选单的可选项。<option>和</option>间可包括事件(见下节)和选单的显示文本。<option> 的属性为可选, 其中 value 属性用来提供值, 当选到该项后, 将该值赋给<select> 的 name 属性所指定的变量。onpick 属性, 用来指定用户选到该项并按 ACCEPT 键后所打开页面的 URL。

test5.wml

```

-----
<?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="card0" ordered="false">
<p>
Please select a city...
<select title="Cities List" name="city">
<option title="Beijing" value="c1">Beijing</option>
<option title="Shanghai" value="c2">Shanghai</option>
<option title="Hongkong" value="c3">Hongkong</option>
</select>
Please select columns...
<select title="Column List" multiple="true" name="col">
<option title="Hot News" value="11">News</option>
<option title="Cool Sports" value="12">Sports</option>
<option title="Pop Enter,,," value="13">Entertainment</option>
</select>
</p>
</card>
</wml>

```

WML 学习（六）

事件

WML 的事件基本上分为两大类，一类是键盘(包括软硬按钮)输入事件，用<do>标签来处理，另一类是相关页面内部的事件，用<onevent>标签来处理。

<do>的语法如下：<do type="type" label="label" name="name" optional="false|true">任务</do>，任务就是以前提到的四种任务。<do>的属性中，type 是必选的，其他为可选。

*label 属性，指定了软按钮在屏幕上的显示文本。目前 type 属性为 delete, help, prev 时该属性无效。

*name 属性，为<do>取个名字，同一的 CARD 里的<do>不能重名。如果 CARD 级的<do>和 DECK 级的<do>同名，则覆盖 DECK 级的<do>。

*optional 属性，指定手机是不是可以忽略这个事件，默认值是 false。

*type 属性，指定触发的事件，具体如下：

type 值	触发原因
accept	调用 ACCEPT 按钮机制
delete	调用 DELETE 按钮机制
help	调用 HELP 按钮机制
options	调用选择按钮机制
prev	调用 PREV 按钮机制
reset	调用清除和重新设定手机状态时的 RESET 机制(目前不支持)

unknown 调用 unknown 机制, 等于 type=""(目前不支持)

vnd.co-type 调用厂商特定的机制(目前不支持)

X-*, x-* 供以后使用(不保留)(目前不支持)

test6.wml

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org
/DTD/wml_1.1.xml">
<wml>
<head>
<meta http-equiv="Cache-Control" content="max-age=0"/>
</head>
<card id="card0" ordered="false">
<do type="accept" label="InputName" name="do1">
<go href="#card01"/>
</do>
<p>
NAME:<input name="userName" title="User Name" type="text" format="*M" emptyo
k="false" maxlength="12"/>
</p>
</card>
<card id="card01">
<p>
You name is $(userName:noesc).
</p>
</card>
</wml>
```

<onevent>的语法如下, <onevent type="type">任务</onevent>, 必选属性 type 的取值如下:

type 值 如果用户执行了以下操作就执行任务

onpick 用户选择或不选一个<option>项时。

onenterforward 用户使用<go>任务到达一个 CARD 时。

onenterbackward 用户使用<prev>任务返回到前面的 CARD 时, 或者按 BACK 按钮时。

ontimer 当<timer>过期时。

test7.wml

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org
/DTD/wml_1.1.xml">
```

```

<wml>
<head>
<meta http-equiv="Cache-Control" content="max-age=0"/>
</head>
<card id="card0" ordered="false">
<p>
Please select a city...
<select title="Cities List" name="city">
<option title="Beijing" value="Beijing">
<onevent type="onpick">
<go href="#card01"/>
</onevent>Beijing</option>
<option title="Shanghai" value="Shanghai" onpick="#card01">Shanghai</option>

<option title="Hongkong" value="Hongkong" onpick="#card01">Hongkong</option>

</select>
</p>
</card>
<card id="card01">
<p>
You are Living in $(city:noesc)
</p>
</card>
</wml>

```

`<timer/>`可以用来在用户不进行任何操作的一段时间后，自动执行一个任务，任何激活 CARD 页面的任务和用户操作都会启动`<timer/>`，而任务进行时，`<timer/>`就停止。每个 CARD 只能有一个`<timer/>`，一个`<timer/>`只能触发一个任务。语法如下：`<timer name="variable" value="value"/>`，其中 `name` 为可选属性，指定为一个变量名，当退出该 CARD 时，该变量存储此时定时器的值，当定时器超时时，手机将该变量设为 0；`value` 为必选属性，用来设置定时器的定时值，最小单位为 0.1 秒。

test8.wml

```

-----
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org
/DTD/wml_1.1.xml">
<wml>
<head>
<meta http-equiv="Cache-Control" content="max-age=0"/>
</head>

```

```

<card id="card1" ontimer="#card2">
<timer name="time1" value="50"/>
<p align="center">
After 5s, goto card2
</p>
</card>
<card id="card2">
<onevent type="ontimer">
<go href="#card1"/>
</onevent>
<timer name="time2" value="50"/>
<p align="center">
Here is card2!
</p>
</card>
</wml>

```

再提一下，<onevent> <timer> <do>三者必须按以上顺序写。

此外在 DECK 级还可以加入<template>，用来将事件捆绑在 DECK 级上，语法如下：

```

<template onenterforward="url" onenterbackward="url" ontimer="url">
<do>或者<onevent>
</template>

```

test9.wml

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org
/DTD/wml_1.1.xml">
<wml>
<head>
<meta http-equiv="Cache-Control" content="max-age=0"/>
</head>
<template>
<do type="accept" label="deckPress" name="do1">
<go href="#card01"/>
</do>
</template>
<card id="card0" ordered="false">
<do type="accept" label="cardPress" name="do1">
<go href="#card02"/>
</do>
<p>

```

Press ACCEPT...

```

</p>
</card>
<card id="card01">
<p>
Here is card01
</p>
</card>
<card id="card02">
<p>
Here is card02
</p>
</card>
</wml>

```

WML 学习 (七)

CGI 编程

1)在 WEB 服务器上添加 WML 的 MIME 类型

对于 IIS4, 可在其管理器里的站点属性中加入新的 MIME 类型, 后缀.wml 和 MIME 类型 text/vnd.wap.wml。

对于 PWS, 可修改注册表, 先在 HKEY_CLASSES_ROOT 层加入主键.wml, 再加入串值 Content Type 为 text/vnd.wap.wml, 然后在 HKEY_LOCAL_MACHINE\Software\CLASSES\MIME\Database\Content Type\中加入主键 text/vnd.wap.wml, 再加入串值 Extension 为.wml。

以前提到的 HDML 也用类似的方法。

2)以 http 方式访问 wml 文件

在手机浏览器的 URL 栏里输入 http://localhost/test.wml, 就可以访问 WML 文件了。

3)CGI 设计

和普通 CGI 变成的方式差不多, 只是返回结果时, 先要输出 Content-Type 为 text/vnd.wap.wml, 然后再输出 WML 内容。

test10.wml

```

-----
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTT/wml_1.1.xml">
<wml>
<head>
<meta http-equiv="Cache-Control" content="max-age=0"/>

```

```

</head>
<card id="card0">
<do type="accept" label="Input Name" name="do1">
<go href="http://localhost/test.asp" method="post">
<postfield name="uid" value="123456"/>
<postfield name="uname" value="\$(userName:esc)"/>
</go>
</do>
<p>
NAME:<input name="userName" title="User Name" type="text" format="*M" emptyok="false" maxlength="12"/>
</p>
</card>
test10.asp
-----
<%
uid=Request.Form("uid")
userName=Request.Form("uname")
Response.ContentType="text/vnd.wap.wml"
%>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org
/DTD/wml_1.1.xml">
<wml>
<card>
<p>
USERNAME:<%=userName%>
USER__ID:<%=uid%>
</p>
</card>
</wml>

```

WAP 网站服务器配置

设置 APACHE WEB SERVER

不管是 NT 还是 UNIX 或 LINUX ， 都是修改 Apache 安装目录下的 conf/mime.types 文件
在该文件中增加以下内容：

text/vnd.wap.wml .wml

image/vnd.wap.wbmp .wbmp

application/vnd.wap.wmlc .wmlc

text/vnd.wap.wmls .wmls

application/vnd.wap.wmlsc .wmlsc

存盘

重新启动 APACHE WEB SERVER 即可。

基于 WINDOW NT 平台的 IIS 设置 WAP SERVER

只要按照以下的简单设置，就可以让 NT IIS 成为 WAP 服务器，您可以撰写 WML 网页或者开发各种服务。

设定方式：(先确定 NT Server 已经安装 IIS 了服务)

执行 NT 操作系统下的【开始】/【程序】/【Windows NT Option Pack】/【IIS 4.0】/Internet 服务管理员。

在 IIS 服务管理员窗口内，点选主机电脑名称后，按鼠标右键，点选“属性”选项；

在属性窗下部有一个文件类型按钮，单击此按钮，会出现文件类型界面；

单击‘新增类型’按钮，然后在相关的扩展名栏中填写 .wml ,在内容类型 (MIME) 栏中填写 text/vnd.wap.wml

单击‘确定’按钮

重复以上步骤将以下 WML 类型新增至 MIME 内： 相关的扩展名

内容类型(MIME)

.wml text/vnd.wap.wml

.wmlc application/vnd.wap.wmlc

.wmls text/vnd.wap.wmlscript

.wmlsc application/vnd.wap.wmlscriptc

.wbmp image/vnd.wap.wbmp

北京网站建设公司 网站建设报价 手机网站建设 英文外贸网站建设 flash 网站制作

网站设计/网站设计公司 网站建设入门知识

网络营销/网站推广 软文 外贸网络营销/外贸 seo 外贸网络推广/网络营销

Wap 网站空间 电子邮件营销 [外贸 seo/英文 seo](#)