DISCIPLINE SPECIFIC 4: WIRELESS APPLICATION PROTOCOL

Semester : VI **Course Code : 16UCS504**

Syllabus

UNIT II:

The Wireless Markup Language: Overview – The WML Document Model – WML Authoring – URLs Identify Content – Markup Basics – WML – Basics – Basic Content – Events, Tasks and Bindings.

UNIT II

1.The Wireless Markup Language: Overview

The topmost layer in the WAP (Wireless Application Protocol) architecture is made up of WAE (Wireless Application Environment), which consists of WML and WML scripting language.

- WML stands for **W**ireless **M**arkup **L**anguage
- WML is an application of XML, which is defined in a document-type definition.
- WML is based on HDML and is modified so that it can be compared with HTML.
- WML takes care of the small screen and the low bandwidth of transmission.
- WML is the markup language defined in the WAP specification.
- WAP sites are written in WML, while web sites are written in HTML.
- WML is very similar to HTML. Both of them use tags and are written in plain text format.
- WML files have the extension ".wml". The MIME type of WML is "text/vnd.wap.wml".
- WML supports client-side scripting. The scripting language supported is called WMLScript.

WML Versions:

WAP Forum has released a latest version WAP 2.0. The markup language defined in WAP 2.0 is XHTML Mobile Profile (MP). The WML MP is a subset of the XHTML. A style sheet called WCSS (WAP CSS) has been introduced along with XHTML MP. The WCSS is a subset of the CSS2.

Most of the new mobile phone models released are WAP 2.0-enabled. Because WAP 2.0 is backward compatible to WAP 1.x, WAP 2.0-enabled mobile devices can display both XHTML MP and WML documents.

WML 1.x is an earlier technology. However, that does not mean it is of no use, since a lot of wireless devices that only supports WML 1.x are still being used. Latest version of WML is 2.0 and it is created for backward compatibility purposes. So WAP site developers need not to worry about WML 2.0.

WML Decks and Cards:

A main difference between HTML and WML is that the basic unit of navigation in HTML is a page, while that in WML is a card. A WML file can contain multiple cards and they form a deck.When a WML page is accessed from a mobile phone, all the cards in the page are downloaded from the WAP server. So if the user goes to another card of the same deck, the mobile browser does not have to send any requests to the server since the file that contains the deck is already stored in the wireless device.You can put links, text, images, input fields, option boxes and many other elements in a card.

2.WML Program Structure/WML Document Model:

A WML program is typically divided into two parts: the document prolog and the body. Consider the following code:

Following is the basic structure of a WML program:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<card id="one" title="First Card">
<p>
This is the first card in the deck
</p>
</card>
<card id="two" title="Second Card">
<p>
Ths is the second card in the deck
</p>
</card>
</wml>
```

WML Document Prolog:

The first line of this text says that this is an XML document and the version is 1.0. The second line selects the document type and gives the URL of the document type definition (DTD). The DTD referenced is defined in WAP 1.2, but this header changes with the

versions of the WML. The header must be copied exactly so that the tool kits automatically generate this prolog.

WML Document Body:

The body is enclosed within a <wml> </wml> tag pair. The body of a WML document can consist of one or more of the following:

- Deck
- Card
- Content to be shown
- Navigation instructions

3.WML AUTHORING

WML documents are plain text files which you can create with any editor or word processor capable of saving files as text. More WML editor d offer various aids such as syntax assistance and color coding.

Static decks however created will take you only so for, because WML is not a programming language. To develop true wireless applications you must employ dynamic content generation using some sort of server side technique. Most mechanisms used for dynamic web programming such as CGI programs or servlets can be used to generate dynamic WML.

Software Development Kits

A **software development kit** (**SDK** or devkit) is typically a set of **software development** tools that allows the creation of applications for a certain **software** package, **software** framework, hardware platform, computer system, video game console, operating system, or similar **development** platform

MobileDev is the first Wireless Development Environment (WDE) specifically for WAP Internet applications. Its innovative open-ended development model integrates a graphical application mapper with a wizard interface and a rich tool set. Versatile enough to satisfy the most demanding professional, MobileDev supports WAP technologies like WML, HDML, Microsoft Active Server Pages (ASP), Perl and Java Server Pages (JSP).

Lloop- an open source software toolkit for creating community portals, codenamed "lloop" has extensive support for WML, the markup language used by WAP phones and other wireless devices. "lloop" consists of a server platform and a collection of ready-for-use ASP-like scripts that produce dynamic content for display on a variety of devices, such as

desktop PC's, WAP Phones, and the Palm VII handheld. The entire project is based upon Open Source technology and is licensed under the GNU Public License.

WAPPage 2.0 is a WSIWYG development tool for WAP (Wireless Application Protocol) Sites. WAPPage 2.0 allows a developer to edit, compile and integrate WML (Wireless Markup Language) Pages. WAPPage 2.0 supports WML 1.1.

Server Setup

Configure Apache Web Server for WAP:

Assuming you have Apache Web server installed on your machine. So now we will tell you how to enable WAP functionality in your Apache web server.

So locate Apache's file httpd.conf which is usually in /etc/httpd/conf, and add the following lines to the file and restart the server:

AddType text/vnd.wap.wml .wml

AddType text/vnd.wap.wmlscript .wmls

AddType application/vnd.wap.wmlc .wmlc

AddType application/vnd.wap.wmlscriptc .wmlsc

AddType image/vnd.wap.wbmp .wbmp

In dynamic applications, the MIME type must be set on the fly, whereas in static WAP applications the web server must be configured appropriately.

Configure Microsoft IIS for WAP:

To configure a Microsoft IIS server to deliver WAP content, you need to perform the following:

Open the Internet Service Manager console and expand the tree to view your Web site entry. You can add the WAP MIME types to a whole server or individual directories.

Open the Properties dialog box by right-clicking the appropriate server or directory, then choose Properties from the menu.

From the Properties dialog, choose the HTTP Headers tab, then select the File Types button at the bottom right.

WAP Media Files

Extension	Media Types
WML files (plain text) (.wml)	text/vnd.wap.wml
WML files (compiled) (.wmlc)	application/vnd.wap.wmlc
WML files (plain text) (.wml)	text/vnd.wap.wml

WMLScript files (compiled) (.wmlsc)	application/vnd.wap.wmlscriptc
WMLScript files (plain text) (.wmls)	text/vnd.wap.wmlscript

4.URLs Identify Content

URL is the abbreviation of **U**niform **R**esource **L**ocator and is defined as the global address of documents and other resources on the World Wide Web.

What Are the Parts of a URL?

The first part of the URL is called a protocol identifier and it indicates what protocol to use, and the second part is called a resource name and it specifies the IP address or the domain name where the resource is located. The protocol identifier and the resource name are separated by a colon and two forward slashes.

http://www.webopedia.com

For the **URL** http://**example**.com , the protocol identifier is http . Resource name: For the **URL** http://**example**.com , the resource name is **example**.com .

Absolute URL

An absolute URL typically takes the following form:

```
protocol://domain/path
```

The protocol is usually http://, but can also be https://, ftp://, gopher://, or file://.

The domain is the name of the website.

Absolute URL requires you to place the entire address on the page that you link to. An example of an absolute URL would look like this:

<a href = http://www.example.com/xyz.html>

Relative URL

The relative URL, on the other hand, does not use the full address. It assumes that the page you type in is on the same site. An example of a relative URL would look like this:

<a href = "/xyz.html">

5.Markup Basics

Simple WML Document

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
```

```
<!-- This will be assumed as a comment -->
<wml>
<card id="one" title="First Card">
<p>
This is the first card in the deck
</p>
</card>
<card id="two" title="Second Card">
<p>
Ths is the second card in the deck
</p>
</card>
</wml>
```

WML - Elements

WML is defined by a set of elements that specify all markup and structural information for a WML deck.

Elements are identified by tags, which are each enclosed in a pair of angle brackets.

Elements have one of the following two structures:

- <tag> content </tag> : This form is identical to HTML.
- <tag />: This is used when an element cannot contain visible content or is empty, such as a line break. WML document's prolog part does not have any element which has closing element.
- Deck & Card Elements

WML Elements	**Purpose**
<!-->	Defines a WML comment
<wml>	Defines a WML deck (WML root)
<head>	Defines head information
<meta>	Defines meta information
<card>	Defines a card in a deck
<access>	Defines information about the access control of a deck

<template>	Defines a code template for all the cards in a deck

- Text Elements

WML Elements	Purpose
 	Defines a line break
<p>	Defines a paragraph
<table>	Defines a table
<td>	Defines a table cell (table data)
<tr>	Defines a table row
<pre>	Defines preformatted text

- Text Formatting Tags

WML Elements	Purpose
<b>	Defines bold text
<big>	Defines big text
	Defines emphasized text
<i>	Defines italic text
<small>	Defines small text
	Defines strong text
<u>	Defines underlined text

- Image Elements

WML Elements	Purpose
<img>	Defines an image

- Anchor Elements

WML Elements	Purpose
<a>	Defines an anchor
<anchor>	Defines an anchor

- Event Elements

WML Elements	Purpose
<do>	Defines a do event handler

<onevent>	Defines an onevent event handler
<postfield>	Defines a postfield event handler
<ontimer>	Defines an ontimer event handler
<onenterforward>	Defines an onenterforward handler
<onenterbackward>	Defines an onenterbackward handler
<onpick>	Defines an onpick event handler

- Task Elements

WML Elements	Purpose
<go>	Represents the action of switching to a new card
<noop>	Says that nothing should be done
<prev>	Represents the action of going back to the previous card
<refresh>	Refreshes some specified card variables.

- Input Elements

WML Elements	Purpose
<input>	Defines an input field
<select>	Defines a select group
<option>	Defines an option in a selectable list
<fieldset>	Defines a set of input fields
<optgroup>	Defines an option group in a selectable list

- Variable Elements

WML Elements	Purpose
<setvar>	Defines and sets a variable
<timer>	Defines a timer

Attributes

Elements can be embellished with attributes which appear as part of the start tag alters the semantics of the elements in some fashion.There are a few core attributes that can be used in nearly every element:

xml:lang-Specifies the language for the element.

Id-Give the item a name that can be referred to later.

Class-Specifies a class name for the element (so they can be grouped).

WML <wml> Tag

The WML <wml> tag is used to define a WML deck and contains cards and other elements of the document.

Attributes:

The <wml> element supports the following attributes:

Attribute	Value	Description
xml:lang	language_code	Sets the language used in the element
class	class data	Sets a class name for the element.
id	element ID	A unique ID for the element.

Example:

Following is the example showing usage of this element:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<card id="one" title="First Card">
<p>
This is the first card in the deck
</p>
</card>
<card id="two" title="Second Card">
<p>
Ths is the second card in the deck
</p>
</card>
</wml>
```

WML <head> Tag

The <head> element in WML is similar to the <head> element in HTML.It marks a place for meta-information about the document to be stored. Meta-information is information about the document itself, rather than its content.If present, this element must be the first thing inside the <wml> element.

Attributes:

The <head> element supports the following attributes:

Attribute	Value	Description
class	class data	Sets a class name for the element.
id	element ID	A unique ID for the element.

Example:

Following is the example showing usage of this element:

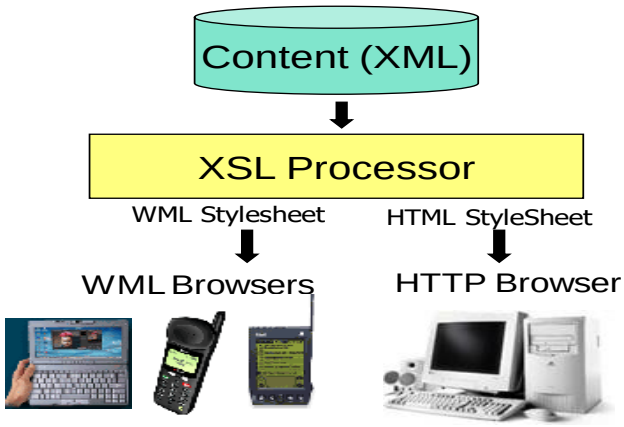
```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<head>
  <access domain="www.tutorialspoint.com"/>
  <meta name="keyword" content="WML"/>
</head>
<wml>
<card id="one" title="First Card">
<p>
This is the first card in the deck
</p>
</card>
<card id="two" title="Second Card">
<p>
Ths is the second card in the deck
</p>
</card>
</wml>
```

6.WML Basics

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

WML: Wireless Markup Language

- Tag-based browsing language:
  - Screen management (text, images)
  - Data input (text, selection lists, etc.)
  - Hyperlinks & navigation support
- Takes into account limited display, navigation capabilities of devices
- XML-based language
  - describes only intent of interaction in an abstract manner
  - presentation depends upon device capabilities
- Cards and Decks
  - document consists of many cards
  - User interactions are split into cards
  - Explicit navigation between cards
  - cards are grouped to decks
  - deck is similar to HTML page, unit of content transmission
- Events, variables and state mgmt



The basic unit is a card. Cards are grouped together into Decks Document ~ Deck (unit of transfer)

All decks must contain

- Document prologue
  - XML & document type declaration
- <WML> element

Must contain one or more cards

WML File Structure

A WML SKELETON

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<head>
……..
…….
</head>
<template>
……
</template>
<card>
……..
</card>
<wml>
```

Each document consists of a Prologue and deck. The deck in turn may contain a head, template and one or more cards; the <head> and <template> elements are optional. The <head> element can contain an <access> element , which specifies who can link to tour deck, and <meta> elements which are used to pass directives to the browser. The <template> element allows you to define event bindings that will apply to all of the cards in the deck.

WML Document Prolog:

The first line of this text says that this is an XML document and the version is 1.0. The second line selects the document type and gives the URL of the document type definition (DTD). The DTD referenced is defined in WAP 1.2, but this header changes with the versions of the WML. The header must be copied exactly so that the tool kits automatically generate this prolog.

The prolog components are not WML elements and they should not be closed, i.e. you should not give them an end tag or finish them with />.

WML Document Body:

The body is enclosed within a <wml> </wml> tag pair. The body of a WML document can consist of one or more of the following:

- Deck
- Card
- Content to be shown
- Navigation instructions

WML <wml> Tag

The WML <wml> tag is used to define a WML deck and contains cards and other elements of the document.

The <wml> element serves a purpose much like the <html> element does for HTML pages.

Attributes:

The <wml> element supports the following attributes:

Attribute	Value	Description
xml:lang	language_code	Sets the language used in the element
class	class data	Sets a class name for the element.
id	element ID	A unique ID for the element.

Example:

Following is the example showing usage of this element:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<card id="one" title="First Card">
<p>
This is the first card in the deck
</p>
</card>
<card id="two" title="Second Card">
<p>
Ths is the second card in the deck
</p>
</card>
</wml>
```

WML <head> Tag

The <head> element in WML is similar to the <head> element in HTML.

It marks a place for meta-information about the document to be stored. Meta-information is information about the document itself, rather than its content.

Attributes:

The <head> element supports the following attributes:

Attribute	Value	Description
class	class data	Sets a class name for the element.
id	element ID	A unique ID for the element.

Example:

Following is the example showing usage of this element:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<head>
  <access domain="www.tutorialspoint.com"/>
```

```
  <meta name="keyword" content="WML"/>
</head>
<wml>
<card id="one" title="First Card">
<p>
This is the first card in the deck
</p>
</card>
<card id="two" title="Second Card">
<p>
Ths is the second card in the deck
</p>
</card>
</wml>
```

Card Element

The <card> element encloses a WML card within a deck. In addition, text and graphics enclosed within <p> elements, it may also contain a number of event bindings

Attributes:

The <card> element supports the following attributes:

Attribute	Value	Description
title	cdata	Gives a title to this card. This title is displayed in some way by the browser when the card is visible.
newcontext	• true • false	Specifies that when this card is entered, the browser context should be cleared.
ordered	• true • false	Provides a hint to the browser about how the card is organized. Set it to true if the card consists of a number of separate fields that should be dealt with in the order they appear in the card. Set it to false if the card contains optional fields or may be filled in out of order.
onenterforward	URL	Occurs when the user navigates into a card using a "go" task
onenterbackward	URL	Occurs when the user navigates into a card using a "prev" task
ontimer	URL	Occurs when a "timer" expires
xml:lang	language_code	Sets the language used in the element.
class	cdata	Sets a class name for the element.
id	element_ID	A unique ID for the element.

Example:

Following is the example showing usage of this element:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
```

```
<card id="one" title="First Card">
<p>
This is the first card in the deck
</p>
</card>

<card id="two" title="Second Card">
<p>
Ths is the second card in the deck
</p>
</card>

</wml>
```

WML - Comments

WML comments use the same format as HTML comments and use the following syntax:

```
<!-- This will be assumed as a comment -->
```

A multiline comment can be given as follows:

```
<!-- This is a multi-line
    comment -->
```

7. Basic Content

Text

Because WML Browser need not support graphics, your content will primarily be text oriented.

Special Characters

To include special characters you must resort to character entities.

WML - Entities

WML entities are to represent symbols that either can't easily be typed in or that have a special meaning in WML.

For example, if you put a < character into your text normally, the browser thinks it's the start of a tag; the browser then complains when it can't find the matching > character to end the tag.

Character Entities

Result	Description	Entity Name	Entity Number
&	ampersand	&amp;	&#38;
'	apostrophe	&apos;	&#39;
>	greater-than	&gt;	&#62;
<	less-than	&lt;	&#60;
	non-breaking space		
"	quotation mark	&quot;	&#34;
	soft hyphen	&shy;	&#173;

Text Formatting Tags/Text Emphasis

WML Elements	Purpose
<b>	Defines bold text
<big>	Defines big text
	Defines emphasized text
<i>	Defines italic text
<small>	Defines small text
	Defines strong text
<u>	Defines underlined text

Examples

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

<wml>
<card title="Formatting">
```

```
<p>
 <em>emphasized text</em><br/>
 <strong>strong text</strong><br/>
 <b>bold text</b><br/>
 <i>italic text</i><br/>
 <u>underlined text</u><br/>
 <big>big text</big><br/>
 <small>small text</small>
</p>
</card>
</wml>
```

Text Elements/Text Layout

WML Elements	Purpose
 	Defines a line break
<p>	Defines a paragraph
<table>	Defines a table
<td>	Defines a table cell (table data)
<tr>	Defines a table row
<pre>	Defines preformatted text

WML <p> Tag

The <p> element defines a paragraph of text and WAP browsers always render a paragraph in a new line.

A <p> element is required to define any text, image or a table in WML.

Attributes:

The <p> element supports the following attributes:

Attribute	Value	Description
align	• left • right • center	This is used to change the horizontal alignment of a paragraph.
mode	• wrap	Sets whether a paragraph should wrap lines or not.

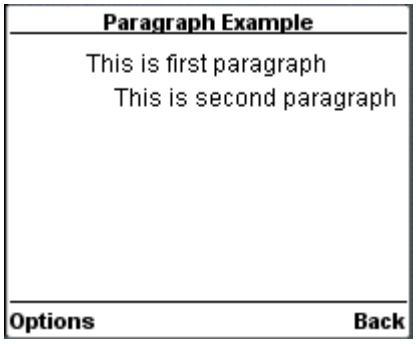
	• nowrap	
xml:lang	language_code	Sets the language used in the element
class	class data	Sets a class name for the element.
id	element ID	A unique ID for the element.

Example:

Following is the example showing usage of element.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<card title="Paragraph Example">
<p align="center">
This is first  paragraph
</p>
<p align="right">
This is second  paragraph
</p>
</card>

</wml>
```

This will produce the following result:



Align attribute

The align attribute specifies the alignment of the text within a paragraph.

Example

This is some text in a paragraph.

Example

```
<card>
<p align=”left”>
Left
</p>
<p align=”right”>
Right
</p>
<p>
Back to Left
</p>
```

Mode Attribute

The mode attribute controls the procedure used to obtain a response from the subject.

Syntax

```
/ mode = responsemode
```

Example

```
<p mode=”nowrap”>
‘tis better to have loved and lost,
<br/>
Than never to have loved at all.</p>
```

**WML
 Tag**

The <br /> element defines a line break and almost all WAP browsers support a line break tag.

Attributes:The <br /> element supports the following attributes:

Attribute	Value	Description
xml:lang	language_code	Sets the language used in the element
class	class data	Sets a class name for the element.
id	element ID	A unique ID for the element.

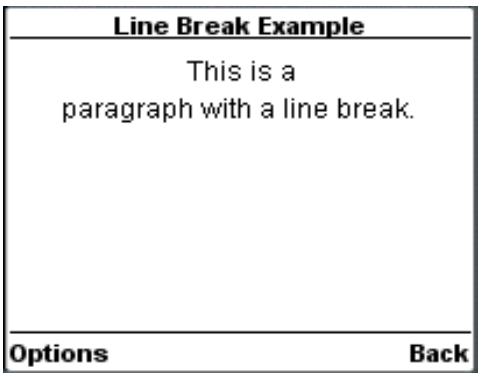
Example:

Following is the example showing usage of <br /> element.

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

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<card title="Line Break Example">
<p align="center">
This is a <br /> paragraph with a line break.
</p></card></wml>
```

result



WML <pre> Tag

The <pre> element is used to specify preformatted text in WML. Preformatted text is text of which the format follows the way it is typed in the WML document.

This tag preserves all the white spaces enclosed inside this tag. Make sure you are not putting this tag inside ...

Attributes:

The <pre> element supports following attributes:

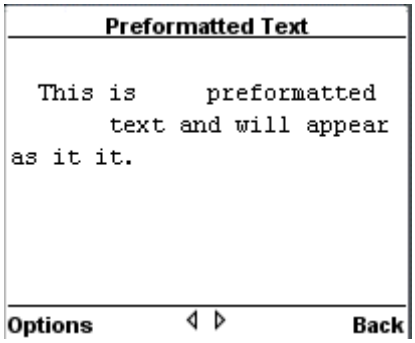
Attribute	Value	Description
xml:lang	language_code	Sets the language used in the element
class	class data	Sets a class name for the element.
id	element ID	A unique ID for the element.

Example:

Following is the example showing usage of <pre> element.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">

<wml>

<card title="Preformatted Text">
<pre>
  This is     preformatted
       text and will appear
 as it it.
</pre>
</card>

</wml>
```

This will produce the following result:



Xml: space attribute

The xml:space attribute can only have two values, "default" and "preserve". Since the "default" **value** in most situations acts as if the attribute is not used, it is seldom used.

8.Events, Tasks and Bindings.

WML - Events

WML language also supports events and you can specify an action to be taken whenever an event occurs. This action could be in terms of WMLScript or simply in terms of WML.

Two types of Events Exist:

- ✓ Intrinsic Event are triggered by WML Elements
  For example , a card can trigger an event are triggered by WML elements
- ✓ User-initiated events are triggered by direct action of the user , such as the selection of an item from menu

WML supports following four event types:

- onenterbackward: This event occurs when the user hits a card by normal backward navigational means. That is, user presses the Back key on a later card and arrives back at this card in the history stack.
- onenterforward: This event occurs when the user hits a card by normal forward navigational means.
- onpick: This is more like an attribute but it is being used like an event. This event occurs when an item of a selection list is selected or deselected.
- ontimer: This event is used to trigger an event after a given time period.

These event names are case sensitive and they must be lowercase.

Trapping Intrinsic Events : WML <onevent> Element:

The <onevent>...</onevent> tags are used to create event handlers. Its usage takes the following form:

```
<onevent type="event_type">
   A task to be performed.
</onevent>
```

You can use either *go, prev* or *refresh* task inside <onevent>...</onevent> tags against an event.

The <onevent> element supports the following attributes:

Attribute	Value	Description
type	• onenterbackward • onenterforward • onpick • ontimer	Defines a type of event occured.
class	class data	Sets a class name for the element.
id	element ID	A unique ID for the element.

Following is the example showing usage of <onevent> element. In this example, whenever you try to go back from second card to first card then **onenterbackward** occurs which moves you to card number three. Copy and paste this program and try to play with it.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">

<wml>
<onevent type="onenterbackward">
  <go href="#card3"/>
</onevent>

<card id="card1" title="Card 1">
<p>
  <anchor>
    <go href="#card2"/>
    Go to card 2
```

```
  </anchor>
</p>
</card>
<card id="card2" title="Card 2">
<p>
  <anchor>
  <prev/>
    Going backwards
  </anchor>
</p>
</card>
<card id="card3" title="Card 3">
<p>
Hello World!
</p>
</card>
</wml>
```

Trapping user initiated Events- The do Element

The <do> tag can be used to activate a task when the user clicks on a word/phrase on the screen.

Attributes:

This element supports the following attributes:

Attribute	Value	Description
name	text	Sets a name for the <do> element.
label	string	Sets a label for the <do> element.
type	• accept • prev • help • reset • options	Defines the type of the <do> element

	• delete • unknown • x-* • vnd.*	
value	number	Specifies the timer after which timer will be expired. Timeouts are specified in units of a tenth of a second.
class	class_data	Sets a class name for the element.
id	element ID	A unique ID for the element.

Example:

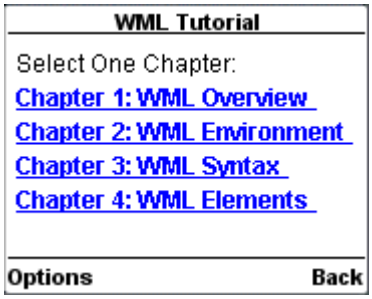
Following is the example showing usage of <do> element along with <go> element.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">

<wml>
  <template>
    <do name="main_menu" type="accept" label="Chapters">
      <go href="chapters"/>
    </do>
    <do name="menu_1" type="accept" label="Chapter 1">

      <go href="#chapter1"/>
    </do>
    <do name="menu_2" type="accept" label="Chapter 2">
      <go href="#chapter2"/>
    </do>
    <do name="menu_3" type="accept" label="Chapter 3">

      <go href="#chapter3"/>
    </do>
```

```
  <do name="menu_4" type="accept" label="Chapter 4">
    <go href="#chapter4"/>
  </do>
</template>

<card id="chapters" title="WML Tutorial">
  <p>
    Select One Chapter:<br/>
    <anchor>
      <go href="#chapter1"/>

      Chapter 1: WML Overview
    </anchor><br />

    <anchor>
      <go href="#chapter2"/>
      Chapter 2: WML Environment
    </anchor><br />

    <anchor>
      <go href="#chapter3"/>
      Chapter 3: WML Syntax
    </anchor><br />

    <anchor>
      <go href="#chapter4"/>

      Chapter 4: WML Elements
    </anchor><br />
  </p>
</card>

<card id="chapter1" title="WML Tutorial Ch1">
```

```
  <p>

    <em>Chapter 1: WML Introduction</em><br/>
    ...
  </p>
 </card>

 <card id="chapter2" title="WML Tutorial Ch2">

  <p>
    <em>Chapter 2: WML Environment</em><br/>
    ...
  </p>
 </card>

 <card id="chapter3" title="WML Tutorial Ch3">
  <p>
    <em>Chapter 3: WML Syntax</em><br/>
    ...
  </p>
 </card>

 <card id="chapter4" title="WML Tutorial Ch4">
  <p>
    <em>Chapter 4: WML Elements</em><br/>
    ...
  </p>

 </card>
</wml>
```

This will produce the following menu and now you can navigate through all the chapters:



WML - Tasks

A WML task is an element that specifies an action to be performed by the browser, rather than something to be displayed. For example, the action of changing to a new card is represented by a <go> task element, and the action of returning to the previous card visited is represented by a <prev> task element. Task elements encapsulate all the information required to perform the action.

WML provides following four elements to handle four WML tasks called go task, pre task, refresh task and noop taks.

Forward Navigation :The <go> Task:

As the name suggests, the <go> task represents the action of going to a new card.

The <go> element supports the following attributes:

Attribute	Value	Description
href	URL	Gives the URL of the new card. Relative URLs are resolved relative to the current card
method	• get • post	Specifies the method that should be used to fetch the deck. This must be one of the values get or post, corresponding to the GET and POST methods of HTTP. When using **method="get"**, the data is sent as an request with ? data appended to the url. The method has a disadvantage, that it can be used only for a limited amount of data, and if you send sensitive information it will be displayed on the screen and saved in the web server's logs. So do not use this method if you are sending password etc. With **method="post"**, the data is sent as an request with

		the data sent in the body of the request. This method has no limit, and sensitive information is not visible in the URL
sendreferer	• true • false	If set to true, the browser sends the URL of the current deck along with the request. This URL is sent as a relative URL if possible. The purpose of this is to allow servers to perform simple access control on decks, based on which decks are linking to them. For example, using HTTP, this attribute is sent in the HTTP Referer header.
accept-charset	charset_list	Specifies a comma- or space-separated list of character sets that can encode data sent to the server in a POST request. The default value is "unknown".
class	class data	Sets a class name for the element.
id	element ID	A unique ID for the element.

Following is the example showing usage of <go> element.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">

<wml>

<card title="GO Element">
<p>
  <anchor>
     Chapter 2 : <go href="chapter2.wml"/>
  </anchor>
</p>
</card>
</wml>
```

Backward Navigation- The prev Element

The <prev> task represents the action of returning to the previously visited card on the history stack. When this action is performed, the top entry is removed from the history stack, and that card is displayed again, after any <setvar> variable assignments in the <prev> task have taken effect.

Attributes:

This element supports the following attributes:

Attribute	Value	Description
class	class data	Sets a class name for the element.
id	element ID	A unique ID for the element.

Example:

Following is the example showing usage of <prev> element.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">

<wml>

<card title="Prev Element">
<p>
  <anchor>
      Previous Page :<prev/>
  </anchor>
</p>
</card>
</wml>
```

Do Nothing- The noop Element

he purpose of the <noop> task is to do nothing (no operation).

The only real use for this task is in connection with templates

Attributes:

This element supports the following attributes:

Attribute	Value	Description
class	class data	Sets a class name for the element.
id	element ID	A unique ID for the element.

Example:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">

<wml>

<card title="Noop Element">
<p>
  <do type="prev" label="Back">
     <noop/>
  </do>
</p>
</card>

</wml>
```

Update the context- The refresh Element

The <refresh> task is the simplest task that actually does something. Its effect is simply to perform the variable assignments specified by its <setvar> elements, then redisplay the current card with the new values. The <go> and <prev> tasks perform the same action just before displaying the new card.

The <refresh> task is most often used to perform some sort of "reset" action on the card.

Attributes:

This element supports the following attributes:

Attribute	Value	Description
class	class data	Sets a class name for the element.
id	element ID	A unique ID for the element.

Example:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">

<wml>

<card title="Referesh Element">
<p>
  <anchor>
      Refresh this page:
       <go href="test.wml"/>
       <refresh>
         <setvar name="x" value="100"/>
       </refresh>
  </anchor>
</p>
</card>

</wml>
```

WML <template> Tag

The <template> is used to apply <do> and <onevent> elements to all cards in a deck. This element defines a template for all the cards in a deck and the code in the <template> tag is added to each card in the deck.You can override a <do> element of a template by defining another <do> element with the same *name* attribute value in a WML card.

Attributes:

The <template> element supports the following attributes:

Attribute	Value	Description
onenterbackward	URL	Occurs when the user navigates into a card using a "prev" task
onenterforward	URL	Occurs when the user navigates into a card using a "go" task
ontimer	URL	Occurs when the "timer" expires
class	class data	Sets a class name for the element.
id	element ID	A unique ID for the element.

Example:

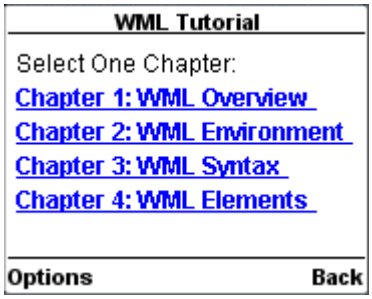
Following is the example showing usage of this element:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">

<wml>
  <template>
    <do name="main_menu" type="accept" label="Chapters">
      <go href="chapters"/>
    </do>
    <do name="menu_1" type="accept" label="Chapter 1">

      <go href="#chapter1"/>
    </do>
    <do name="menu_2" type="accept" label="Chapter 2">
      <go href="#chapter2"/>
    </do>
    <do name="menu_3" type="accept" label="Chapter 3">
```

```
    <go href="#chapter3"/>
  </do>
  <do name="menu_4" type="accept" label="Chapter 4">
    <go href="#chapter4"/>
  </do>
</template>

<card id="chapters" title="WML Tutorial">
  <p>
    Select One Chapter:<br/>
    <anchor>
      <go href="#chapter1"/>

      Chapter 1: WML Overview
    </anchor><br />

    <anchor>
      <go href="#chapter2"/>
      Chapter 2: WML Environment
    </anchor><br />

    <anchor>
      <go href="#chapter3"/>
      Chapter 3: WML Syntax
    </anchor><br />

    <anchor>
      <go href="#chapter4"/>

      Chapter 4: WML Elements
    </anchor><br />
  </p>
</card>
```

```
<card id="chapter1" title="WML Tutorial Ch1">
  <p>

    <em>Chapter 1: WML Introduction</em><br/>
    ...
  </p>
</card>

<card id="chapter2" title="WML Tutorial Ch2">

  <p>
    <em>Chapter 2: WML Environment</em><br/>
    ...
  </p>
</card>

<card id="chapter3" title="WML Tutorial Ch3">
  <p>
    <em>Chapter 3: WML Syntax</em><br/>
    ...
  </p>
</card>

<card id="chapter4" title="WML Tutorial Ch4">
  <p>
    <em>Chapter 4: WML Elements</em><br/>
    ...
  </p>

</card>
</wml>
```

This will produce the following menu and now you can navigate through all the chapters:


WML Tutorial
Select One Chapter:
Chapter 1: WML Overview
Chapter 2: WML Environment
Chapter 3: WML Syntax
Chapter 4: WML Elements
Options
Back