

WML - TASKS

http://www.tutorialspoint.com/wml/wml_tasks.htm

Copyright © tutorialspoint.com

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 tasks.

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 | <ul style="list-style-type: none">getpost | <p>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.</p> <p>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.</p> <p>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</p> |
| sendreferer | <ul style="list-style-type: none">truefalse | 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>

```

Another example showing how to upload data using Get Method

```

<?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>
    Using Get Method
    <go href="chapter2.wml?x=17&y=42" method="get"/>
  </anchor>
</p>
</card>
</wml>

```

Another example showing how to upload data using <setvar> 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>
    Using setvar:
    <go href="chapter2.wml">
      <setvar name="x" value="17"/>
      <setvar name="y" value="42"/>
    </go>
  </anchor>
</p>
</card>
</wml>

```

Another example showing how to upload data using <postfield> 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>
    Using setvar:
    <go href="chapter2.wml" method="get">
      <postfield name="x" value="17"/>
      <postfield name="y" value="42"/>
    </go>
  </anchor>
</p>
</card>
</wml>

```

The <prev> Task:

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.

If no previous URL exists, specifying <prev> has no effect.

The <prev> 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. |

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>
```

One situation where it can be useful to include variables in a <prev> task is a login page, which prompts for a username and password. In some situations, you may want to clear out the password field when returning to the login card, forcing the user to reenter it. This can be done with a construct such as:

```
<?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>
    <prev>
      <setvar name="password" value=""/>
    </prev>
  </anchor>
</p>
</card>
</wml>
```

The <refresh> Task:

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.

The <refresh> 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. |

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

```
<?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>
```

The <noop> Task:

The purpose of the <noop> task is to do nothing *nooperation*.

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

The <noop> 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. |

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

```
<?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>
```