B336 Advanced Internet Computing

WML Script (2)

Learning Objectives

- Study WMLScript language

How to Call Script Functions from WML

- Script functions can be called from either link element of WML: <go> or <a> e.g.:

```
<do type="accept">
  <go href="http://www.blah.com/wmlscript#calculate($result, $l)">
</do>
```

- In this example, <go> points to a script function, calculate, as the target resource. The script is run in a separate context from the WML card, and script variables are not shared with the WML card
- Control is returned to the card when the script ends
- No return value from script, but it can manipulate WML card variables
- Script functions are not added to the history list

WMLScript Libraries

- To do something useful in WMLScript, need to use the built-in libraries supported by the WAP browser:

  WMLBrowser, Dialogs, Lang, Float, String and URL
- Use these to:
  - alter the state of the browser by setting & changing variables in the WML document
  - Generate simple pop-up dialogs to ask for input, make a selection, or confirm some event
  - Work with strings, floating-point numbers and URLs
  - Find out system properties such as whether or not floats are supported
  - Generate random numbers
  - Terminate scripts in different ways
- To call a library function, use the form libraryname.functionname
- Data types include Boolean, Float, Integer, String, Invalid (if any para is in error, function returns this)
- Parameters are passed by value

WMLScript Libraries: WMLBrowser

- setVar(*name*, *value*) binds the var *name* to *value*
- getVar(*name*) retrieves the value of var *name*
- The refresh( ) function updates the current display

```
 extern function repaint()
{
  var text = WMLBrowser.getVar("text");
  if (text == "Blix")
    WMLBrowser.setVar("text", "");
  else
    WMLBrowser.setVar("text", "Blix");
WMLBrowser.refresh();
}
```

var text toggles between values of "Blix" and nil

```
<card id="card1" title="Intro" onenterforward="repaint.wmls#repaint()"
          ontimer="repaint.wmls#repaint()>
   <timer value="10" />
   <p align="center">
      <br/>
      <big>$text</big>
   </p>
</card>
```

value of text is updated on each ontimer event (ie every 1 second)

WMLScript Libraries: WMLBrowser

- The previous example could be extended with more variables, text, simple ASCII graphics or animation
- By using more than one card and .wbmp images, simple multimedia effects can be produced
- Variables in the WML document can be cleared using the newContext( ), which is like the WML attribute
- To make the browser jump to a new location, use the go(*url*) function. *url* can be absolute or relative to the current document
- prev() jumps to the last document on the history list
- getCurrentCard() fetches the URL of the card being displayed by the browser

WMLScript Libraries: Dialogs

- Use Dialogs to trigger pop-up boxes and get info from the user
- alert(*text*) informs user of event by displaying *text*
- confirm(*text*, *choiceOK*, *choiceNOK*) allows a choice between values *choiceOK* and *choiceNOK*, in response to the question posed in *text.*
- prompt(*text*, *defaultValue*) accepts input in response to text, or *defaultValue* if no input was entered

```
 extern function setname()
{
  var result = Dialogs.prompt("Who is the author?");
  WMLBrowser.setVar("name", result);
  WMLBrowser.go("#author");
}
```

accepts author's name and sets var name accordingly

```
<card id="intro" onenterforward="check.wmls#setname()" />

<card id="author"><p>$(name)</p></card>
```

the first card calls setname

the second card is called by setname to display the name

WMLScript Libraries: Lang

- Useful functions to convert between data types, fetch system values and generate random numbers
- Some WAP browsers can't do floating point numbers: float() returns true if the current browser can
- maxInt() and minInt() return the max and min integers supported by the current browser, respectively
- isInt() checks to see if a given string might be able to be converted to an integer and parseInt() does it:

```
var y = isInt("13.13");     //Returns true
var y = parseInt("13.13");  //Returns "13"
```

- isFloat() and parseFloat() work the same way for floating point numbers

WMLScript Libraries: Lang

- characterSet() returns the MIBEnum value assigned to character sets by the Internet Naming Authority (eg ISO-8859-1 = 4 and shift_JIS = 17)
- abs(*number*), max(*num1*, *num2*) and min(*num1*,*num2*) all do what you expect
- Generate a positive pseudorandom number using random(*value*); the result will be 0 <= result <= *value* Set the seed using seed(*value*)
- abort(*string*) terminates a script with an error message while exit(*value*) terminates normally
- Most browsers ignore the parameters of these functions

WMLScript Libraries: Float

- int(*number*) returns the integer part of the number
- floor(*number)* returns greatest integer <= number
- ceil(*number*) returns smallest integer >= number
- round(*number*) rounds *number* off to closest integer
- sqrt(*number*) returns square root
- pow(*num1*, *num2*) raises *num1* to the power *num2.* If *num1* is negative, *num2* must be an integer
- maxFloat returns the maximum floating point value
- minFloat returns the minimum floating point value

WMLScript Libraries: Float example

```
 extern function Floatcheck()
{
  var a = Float.int("13.5");
  var b = Float.floor("13.5");
  var c = Float.ceil("13.5");
  var d = Float.round("13.5");
  var e = Float.sqrt("13.5");
  var f = Float.minFloat();
  var g = Float.maxFloat();

  WMLBrowser.setVar("a",a);
  WMLBrowser.setVar("b",b);
  WMLBrowser.setVar("c",c);
  WMLBrowser.setVar("d",d);
 }
```

```
<card id=card1" title="FloatCheck">
  <onevent type="onenterforward">
     <go href="floatcheck.wmls#Floatcheck()" />
  </onevent>
  <p>
     <table columns="2" align="LR">
       <tr><td>Float.int("13.5")</tr></td> => $a</td></tr>
       <tr><td>Float.floor("13.5")</tr></td> => $b</td></tr>
       <tr><td>Float.ceil("13.5")</tr></td> => $c</td></tr>
       <tr><td>Float.round("13.5")</tr></td> => $d</td></tr>
       <tr><td>Float.sqrt("2.2")</tr></td> => $e</td></tr>
       <tr><td>Float.minFloat()</tr></td> => $f</td></tr>
       <tr><td>Float.maxFloat()</tr></td> => $g</td></tr>
     </table>
  </p>
</card>
```

WMLScript Libraries: String

- length(*string*) returns its length; isEmpty(*string*) returns True if *string* is empty, False otherwise.
- charAt(*string*, *index*) will access the character at position *index* of *string.* If out of range, an empty string will be returned
- subString(*string*, *startIndex*, *length*) will return *length* number of characters from *string* starting from *startIndex*
- find(*string*, *subString*) returns the index of the first character in *string* that matches *subString* (-1 if none)
- replace(*string*, *oldsubString, newsubString*) swaps all *oldsubStrings* in *string* for *newsubStrings*

WMLScript Libraries: URL

- The URL library is used to check whether a URL is valid, to request a part of a URL, to request content from a URL and for URL escape and unescape chars

URL format `<scheme>://<host>:<port>/<path>;<params>?<query>#<fragment>`

A valid URL `http://www.blix.com:8080/wap/index.wml;3;2?author=peter#name`

- isValid(*url*) returns True if *url* is valid as above
- getScheme, getHost, getPort, getPath, getParameters getQuery and getFragment all fetch their parts of *url*
- When a script is invoked from a WML document, that document is called the *referer*. The URL of the referrer can be had using getReferer( )

WMLScript Libraries: URL

- An absolute URL is a fully instantiated form such as "http://www.blix.com/script.wmls" while a relative URL is a partial form such as "/index.wmls"
- A script has a base URL where it's code is stored and from which relative URLs derive. This can be obtained using the getBase( ) function
- If necessary, a full URL can be computed from a base URL and a relative URL with resolve(*base*, *relative*) eg

```
var absoluteURL = URL.resolve("http://www.blix.com", "index.wml");
```

will return the absolute "http://www.blix.com/index.wml"

WMLScript Libraries: URL example

```
 extern function parseUrl()
{
  WMLBrowser.newContext();
  var base = URL.getBase();
  var aUrl = Dialogs.prompt("Enter URL:", base);

  if (URL.isValid(aUrl))
  {
    var scheme = URL.getScheme(aUrl);
    var host = URL.getHost(aUrl);
    var port = URL.getPort(aUrl);
    var file = URL.getPath(aUrl);
    var para = URL.getParameters(aUrl);
    var query = URL.getQuery(aUrl);
    var frag = URL.getFragment(aUrl);

    WMLBrowser.setVar("scheme", scheme);
    WMLBrowser.setVar("host", host);
    WMLBrowser.setVar("port", port);
    WMLBrowser.setVar("path", file);
    WMLBrowser.setVar("parameters", para);
    WMLBrowser.setVar("query", query);
    WMLBrowser.setVar("frag", frag);
   }
   else
      Dialogs.alert("Invalid URL");

   WMLBrowser.refresh();
 }
```

```
<card id="show"
        onenterforward=parseurl.wmls#parseUrl">
  <p>
    Scheme: $scheme <br/>
    Host: $host <br/>
    Port: $port <br/>
    Path: $path <br/>
    Parameters: $parameters <br/>
    Query: $query <br/>
    Fragment: $frag <br/>
  </p>
</card>
```

the WML document to invoke this script might look like this

WMLScript Libraries: URL example

- The content from a resource can be loaded directly into a string variable using loadString(*url*, *contentType*) Only textual content is supported
- The following script downloads a data file with values separated by semicolons

```
 extern function getData()
{
  WMLBrowser.newContext();

  var absoluteUrl = URL.resolve(URL.getBase(),"data.txt");
  var data = URL.loadString(absoluteUrl,"text/plain");

  var title = String.elementAt(data,0,";");
  var fname = String.elementAt(data,1,";");
  var lname = String.elementAt(data,2,";");

  WML.Browser.setVar("title", title);
  WML.Browser.setVar("fname", fname);
  WML.Browser.setVar("lname", lname);

  WMLBrowser.refresh();
 }
```

this file contains lines of data such as WAP;Peter;Stark

References

Anderson, et.al. *Professional XML*. Wrox Press, 2000. Chapter 14.

And see the official pages

WAP Forum at http://www.wapforum.org
Developer's Forum at http://www.wapdevelopers.org

Vendors also have good pages:

http://www.nokia.com/
http://www.ericsson.se/WAP
http://www.phone.com/