WAP *on the attack*

Internet-enabled mobile phones have finally hit the UK. Here, Gordon Laing sings the praises of using your WAP browser to find the nearest curry house and Will Head has a guide to building WML pages.

Today there are over 300 million mobile phone subscribers in the world which, according to Nokia, is around twice the number of PCs connected to the Internet. By 2003 there'll be an estimated one billion mobile subscribers, which is a market that content providers simply cannot ignore. But until high-speed, third-generation mobile networks arrive, how do you go about delivering Internet-based content to mobile phones with their small mono displays, slow processors, tricky keypads and poor battery life, not to mention their ever-so-slow networks of variable quality? The answer is WAP, the Wireless Application Protocol.

While products such as Nokia's Communicator 9000/9110 have long provided wireless, but oh-so-slow HTML browsers, WAP is designed from the ground up to deliver Internet-based services to challenging mobile environments. It employs the Wireless Markup Language (WML) to describe how content is displayed to the user and how the system should respond to requests.

As far as appearance is concerned, one way to cope with paltry mobile data and small mono screens has been to strip away all the fun stuff. The sad fact is that navigating WML content through a WAP browser today resembles little more than an interactive text-messaging session with a remote server; check Gelon's website for a preview of how a Nokia 7110 displays WAP sites.

Basic text, however, is adequate for delivering news headlines, travel and weather information, sports scores and share prices, along with drilling down into directories and entertainment listings or ticketing, messaging and banking services.

So, what do you need to get up and running with WAP? Well, while you could run a WAP browser on any platform, including Windows, the service is designed for mobile phones. To access WAP content with a mobile, your handset needs a WAP browser, a data-enabled account, a connection to the Internet and the address of a WAP gateway.

From early tests, it would appear that any existing dialup Internet account can be used, although you'll also need the IP address of a suitable WAP gateway to proceed any further; but ISPs are bound to offer a WAP gateway to subscribers at some point.

Orange, Cellnet and Vodafone provide a dialup account to their data subscribers and don't currently charge for accessing their WAP gateways. You only pay for the actual dialup call charge, which costs around 5p per minute. One2One's WAP service is expected soon.

As for the hardware, Nokia's dual-band 7110 with spring-loaded mouthpiece is the first WAP 1.1-compliant mobile phone. It supports the improved 14.4Kbits/sec, circuit-switched mobile data rate (even if no UK network does as yet), and comes with infra-red drivers that talk to Windows notebooks, Windows CE and Psion Series 5. If you can track one down, it's a bargain at £129.99 with a new subscription to Orange, Cellnet or Vodafone – check your dealer for upgrade prices.

Ericsson's forthcoming R320 will be its first WAP mobile, but the gadget fanatics will wait for the R380, where the keypad flips open to reveal that just about the entire front surface of the handset is a touch-sensitive screen. Ericsson has also agreed to use Microsoft's Mobile WAP Explorer for its handsets.

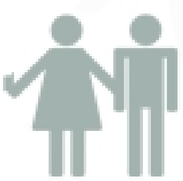
Motorola is about to launch two WAP phones: the tri-band P7389 is essentially an L7089 with Phone.com's WAP Microbrowser, while the P1088 is a bulky smartphone, with huge screen, handwriting recognition, and a Java-compatible HTML browser to go with the WAP side of things.

Siemens' S25 has a WAP browser, but it uses text messaging as delivery; the forthcoming S35 will be a proper WAP 1.1 unit.

You'd be surprised at what the WAP sites have to offer. Starting with the network's own content, you'll find a wealth of useful news services, travel and weather information, entertainment listings and directories.

We were particularly impressed with Orange's services, finding addresses and numbers of some pretty obscure stores in the business directory, and tracking down the precise times of any film at any cinema in the UK.

Cellnet's content is based around its Genie service which, like Orange, supplies loads of decent news, sports and entertainment information. Genie's site also lets you access your Genie-based email on a WAP phone.



Picking up email could end up being a killer application for WAP, and on our travels we found several remote POP3 client services, which worked similar to the way Hotmail and Excite handle POP3 accounts. Room33, and Webcab.de supply email services, while waps.co.uk offer a POP3 client and access to an open gateway for £100 per year.

WAP banking, which would let you check your balance, transfer money and pay bills from your phone, is another killer app and, as we went to press, the UK networks were busy announcing deals: Orange with NatWest, Cellnet with the Halifax, and Vodafone with the Woolwich. The level of exclusivity is unknown, but it's likely that, say, non-Cellnet subscribers would still be able to check their Halifax accounts, but perhaps with a limited range of options – much like using a different bank's cashpoint machine.

SITE ADDRESSES

Viewable on WAP browsers

Waptastic:	http://www.waptastic.com/wml.asp
Webcab:	http://webcab.de/wwe/i.wml

HTML sites

Waptastic:	www.waptastic.com
Webcab.de:	http://webcab.de
WAP forum:	www.wapforum.org
Phone.com:	www.phone.com
Genie:	www.genie.co.uk
Digital Mobility:	www.digimob.com
Gelon:	www.gelon.net
Room33:	www.room33.com
WAPS:	www.waps.co.uk
Wireless Games:	www.wirelessgames.com
Digital Bridges:	www.digitalbridges.com

We're told WAP gaming will be big, but during our tests we couldn't get services from sites such as Wireless Games or Digital Bridges running on our handset. Shopping and auctions could also take off on WAP, with Lastminute.com already present, and free-ads paper *Loot* talking about developing a WAP presence.

Some WAP services require that you dial directly into them, such as Digital Mobility's business services. Pricing was unknown as we went to press, but like subscription-free ISPs, WAP services could make their money from the dialup number alone.

No round-up of sites would be complete without a set of portals, and we've found a pair of crackers: Webcab.de and Waptastic both provide links to a huge range of WAP sites. Favourites we keep going back to are the comprehensive BBC WAP site, and the ever-popular Curryhouse, that finds your closest Indian restaurant.

The future of WAP is far from certain. It could be argued that WAP solves two problems that may not be around in a year: namely slow mobile comms and poor displays. Forthcoming packet-switched and third-generation mobile services claim to offer up to 100Kbits/sec by the end of this year, 400Kbits/sec soon after that, and up to 2Mbits/sec in 2002. Bluetooth connections to PDA-style displays could provide the detailed colour graphics we're used to on PC web browsers. With a fast link and decent screen, you could easily bypass WAP and stick with HTML.

Of course that's all in the future, and it's far from set in stone. Current WAP services may not be visually compelling, but the limited functionality of a mobile handset is adequate for their delivery. In fact, in this day and age of unnecessarily slow, flabby websites, it's quite refreshing to see content stripped down to its bare bones for delivery to modest hardware.

Our initial experiences with WAP have seen it resembling some PC traits you'd rather do without. During early tests with Nokia's 7110 we frequently found sites that crashed the phone to a point that the battery had to be removed. We found that gateways which happily let us browse the world one day, were closed for anything other than internal use the next. There were also many sites that had moved or didn't want to respond, resulting in a 'Connection Failed' message that will become as hated as '404 Not Found'.

In other words, WAP at the time of writing is no different from the first days of the web. There's not a great deal of content, and much of what exists either doesn't yet work properly, or is little more than a placeholder for forthcoming services. WAP, like HTML, is also a maturing standard, and it's unknown how easily phones can be upgraded to newer browsers. In fact with just the Nokia 7110 available at the time of writing, it's uncertain how well WML content will fare on other hardware.

The mobile industry, however, is behind it, and an estimated 70 per cent of all new handsets released by the end of this year will feature WAP capabilities. Since these will also inevitably feature longer battery life, better screens, greater features and more stylish designs, you've really got nothing to lose in choosing WAP the next time you upgrade your mobile.

Turn over for Will Head's step-by-step workshop on how to set up your own WAP pages.

WAP *workshop*

ASK PEOPLE WHAT THEY THINK about WAP and they'll probably tell you that it lets you access the Internet from a mobile phone. While this is true to an extent, buying a mobile phone to access the Internet is like buying a moped to drive across the US. However, if you just want to make quick trips into town, then nipping around on a moped is probably more effective. It's a case of choosing the most suitable device for what you need to do.

WAP pages are created using WML rather than HTML. WML has been designed to provide quick access to information from mobile devices with cramped displays. HTML, on the other hand, has become large and fragmented, with different browser developers extending it in their own ways. To get around the bloat and multiple standards of HTML, WAP uses the simpler WML. This is a dialect of XML and it has a strict structure. In HTML a browser ignores tags it doesn't recognise, but a WAP browser is more inclined to give a non-specific error message. This means you need to test your WAP pages thoroughly before posting them to the server.

The proof is in the pudding, and you can point your WAP-enabled browser to our example site at wap.kerbang.com for a taster

A basic WML page should have the following structure:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "- ↙
//WAPFORUM//DTD WML 1.1//EN" ↙
"http://www.wapforum.org/DTD/↙
wml_1.1.xml">
<wml>
  <template>
  </template>
  <card id="one" title="one">
    ... WML code ...
  </card>
  <card id="two" title="two">
    ... more WML code ...
  </card>
</wml>
```

(*Key:* ↙ *code string continues*)

The first line identifies the file as an XML document and the second specifies that the page has been written for WAP 1.1 browsers (such as the Nokia 7110). The rest of the file must be enclosed inside the <wml> </wml> tags (similar to <html> </html> on a web page). Another major difference is that a WML file consists of small, discreet chunks of information called cards. Each card should contain about a screen's worth of information. The file containing all the cards is referred to as a deck. Finally the <template> section defines common items that will appear on all the cards in a deck.

So if we wanted a simple card to display the 'Hello World!' message it would look like this:

```
<card id="hello" title="Hi">
  <p align="center">
    Hello World!
  </p>
</card>
```

All text that you want to display must be contained within <p> and </p> tags, otherwise you'll get an error. You may start to get annoyed at how strict the language is, but after a while you'll start to wish that HTML was this pedantic. At least with WML, if you follow the rules, your page will display on any appropriate device.

So the next step is to upload it to a webserver. However, after FTPing it to the server you'll find your attempts to access it will probably produce an error. 'But I followed all the rules', I hear you cry! Well you may have followed all the rules correctly but there is one thing missing – MIME types (Multi-purpose Internet Mail Extensions).

MIME types, as you've probably guessed, were originally developed for email packages, but their use has extended to include web servers and clients. MIME types allow a server to tell a browser what type of information it is sending. It also lets a browser decide what to do with the information, a bit like file associations in Windows.

For example, if a browser receives an HTML type it knows it can display the document. If, on the other hand, it receives a ZIP type, it knows to save it to disk and not just dump it in the browser window (well that's the theory anyway). The standard MIME type for HTML documents is 'text/html', for WML pages we need to send the MIME type 'text/vnd.wap.wml'.

One way to solve this problem is to edit the server configuration files and add the correct MIME type. If you've got your own server then this shouldn't be a problem, but most of us haven't. Another option is to talk to your hosting company and see if it is prepared to do this for you, although it may charge an admin fee. If it is not prepared to do this or you don't want to pay then there is another option if your pages are hosted on an Apache server. Create a file called '.htaccess' with the following content:

```
addtype text/vnd.wap.wml wml
```

Place it in the directory you want to put the WML files and whenever a WML file is requested the server will respond with the correct MIME type. You can now go off and happily code WML to your heart's content. However, there is one other gotcha. If you've edited the .htaccess file then the server is unlikely to deliver index.wml as the default document if the user goes to www.yourserver.com/wap, and since typing addresses into a mobile handset is a tedious business then keeping the address as short as possible is paramount.

One work-around for those lucky people who can execute cgi files in any directory is to create a file called index.cgi as follows:

```
#!/usr/local/bin/perl
print "Content- ↙
type:text/vnd.wap.wml\n\n";
print <<END;
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "- ↙
//WAPFORUM//DTD WML 1.1//EN" ↙
"http://www.wapforum.org/DTD/↙
wml_1.1.xml">
<wml>
... WML Code ...
</wml>
END
;
```

Modify the first line so that it points to the perl executable on your server (ask your hosting provider if you're unsure where this is). This script basically delivers the correct MIME type (print "Content-type:text/vnd.wap.wml\n\n";) and then the rest of the WML file. To test this out you'll need to upload your file to the server as ASCII, not binary, and then execute the following command to allow the script to be executed:

```
chmod 755 index.cgi
```

When the user goes to www.yourserver.com/wap then index.cgi will be executed and the correct MIME type and content delivered. You can then link to other WML files in the directory. If you want to see this in action (assuming you have a 7110 or suitable WAP browser) we've set up a working example at: www.kerbang.com/wap.

Although we've got the URL down by nine characters (16 keypresses when entering the characters on a mobile phone) we can make it even shorter. The general trend that is emerging is to use a subdomain called 'wap' for WAP pages. So, for example, wap.kerbang.com would be the address of the WAP site, snipping three keypresses. One way to do this is to set up the subdomain on a separate IP address and serve the WAP pages from there, although this will usually mean paying for another hosting account. The other option is to set up an index.cgi in the root of your webserver. This will redirect a WAP user that enters wap.kerbang.com to your WAP content and a HTML user that enters www.kerbang.com to your HTML page. To do this you need to set up up an alias so wap.yourserver.com points to the same place as www.yourserver.com (you'll need to get your host provider to do this).

Then you'll need the following script in the html root of your webserver (again you need to be able to execute cgi scripts in any directory):

```
#!/usr/local/bin/perl
$host_name = $ENV{'HTTP_HOST'};
if ($host_name =~ ↙
/wap.kerbang.com/) {
# Redirect to ↙
www.kerbang.com/wap
print "Location: ↙
http://www.kerbang.com/wap/\n\n";
} else {
# Redirect to www.kerbang.com/html
print "Location: ↙
http://www.kerbang.com/html/\n\n";
}
```

This example assumes you are keeping your WML file in a directory called 'wap' and your HTML files in a directory called 'html'. It finds out which host name was typed in, ie wap.kerbang.com or www.kerbang.com and redirects the browser to the correct place.

The final icing on the cake would be to create a script that also detects whether a browser can accept WML or HTML content and redirect the user as appropriate. With a little modification the script becomes:

```
#!/usr/local/bin/perl
$host_name = $ENV{'HTTP_HOST'};
$accept_types = ↙
$ENV{'HTTP_ACCEPT'};
if ($host_name =~ ↙
/wap.kerbang.com/) {
# Redirect to www.kerbang.com/wap
print "Location: ↙
http://www.kerbang.com/wap/\n\n";
} else {
if ($accept_types =~ ↙
m|text/vnd.wap.wml|) {
# Redirect to www.kerbang.com/wap
print "Location: ↙
http://www.kerbang.com/wap/\n\n";
} else {
# Redirect to www.kerbang.com/html
print "Location: ↙
http://www.kerbang.com/html/\n\n";
}
}
```

First we check for wap.kerbang.com and then failing that we check to see whether the browser accepts WML content and redirect it appropriately. If both of these tests fail, we resort to simple HTML.

With care it should be possible to set up a server to deliver both HTML and WML content to the appropriate user.

There is more information on the intricacies of WML coding in waptastic.com's series of articles on 'WAP for web developers'. If you need help setting up your server to deliver the content, then one useful resource is the Server Check that can be found in gelon.net's developer section.

We would like to thank Cellnet for lending us a Nokia 7110 for testing purposes. www.btcellnet.co.uk

WILL HEAD

Gelon.net's developer section will test your site to see if your server is correctly set up to handle WAP content