Migration from CDPD to GPRS
Application Development Considerations

White Paper

By Scott McEntire, AT&T Wireless

Revision **1.0**
Revision Date **6/26/01**

Revision History

Date	Revision	Comments
February 19, 2001	0.1	First Draft – Scott McEntire.
March 7, 2001	0.2	Minor revisions from Scott Stern.
March 9, 2001	0.3	Added rough WML application code.
March 20, 2001	0.4	Added several helpful Web sites.
June 18, 2001	1.0	Received approval from Legal and Marcom

Table of Contents

Introduction

On November 30, 2000, AT&T Wireless announced a strategic alliance with NTT DoCoMo in the U.S. and Japan to jointly develop multi-media services based on its current and next-generation high-speed wireless network. It also announced its plans to be among the first North American-based carrier to deploy the global standard UMTS (Universal Mobile Telecommunications System) also known as WCDMA (wideband code division multiple access) networks. These plans help AT&T Wireless to accelerate the deployment of GSM voice and GPRS data networks, bringing us closer and strategically positioned for the 3G network deployments in 2002.

AT&T Wireless is in the process of creating a new, exclusively owned subsidiary to develop multimedia applications for its current network and a new, high-speed wireless network built to global standards for "third-generation" (3G) services such as graphic presentation of data, video, e-mail, high quality music downloads, and streaming audio and video. Both companies will share technical resources and support staffing of the new unit.

From a wireless application development perspective it is important to realize that like the AT&T Wireless CDPD network, GPRS will be an IP packet data network. AT&T Digital PocketNet service applications, as well as standard applications written to the IP stack do not need to be rewritten. HDML applications will need to be re-written in WML to function on devices without an Openwave browser. As new GPRS devices are introduced and our WAP gateway is upgraded with advanced WML language features, developers may want to consider rewriting or creating new applications that fully exploit new language features, multiple devices with disparate browsers, and increases in network speeds. AT&T Wireless is also working with market-leading vendors to develop next-generation wireless devices including data only and voice and data handsets, PDA's, smart-phones, and other wireless devices.

Introduction to GSM and GPRS

GSM: An Overview

GSM is the leading second-generation standard for mobile phones worldwide. It leads both in terms of market share -- about two-thirds of all mobile phones are GSM phones -- and in terms of availability. GSM service is now available in almost all countries (the major exceptions are Japan and Korea). GSM is second generation, or 2G, in that it was first developed in the 1980s during the rollout period of first-generation services. Like all the second-generation standards, it uses exclusively digital techniques for transmission and control. The first-generation standards used analog techniques for transmission.

GSM differs from the third-generation standards in that it is primarily designed to transmit voice signals between telephones, The third-generation standards are designed to transmit a much greater diversity of data streams. The European Telecommunications Standards Institute (ETSI) membership and technical teams built the GSM standard. The Third Generation Partnership Project (3GPP) now owns GSM along with its own successor standards.

GSM works by using a combination of frequency division multiplexing and time division multiplexing to share the radio bandwidth between different phone conversations. These two techniques are called Frequency Division Multiple Access and Time Division Multiple Access, or FDMA and TDMA. GSM also uses the principle of space division multiple access, simply by being a system based on a series of base stations, each covering a limited area. This cellular principal of space division multiple access is common to all three generations of mobile telephone technologies.

FDMA for GSM slices up the available radio spectrum into chunks that are 200KHz wide. These frequency bands are re-used many times over the whole of a network but are not used in adjacent areas to avoid interference. This re-use principle was used for first-generation systems but the radio channels were narrower and carried only one conversation each. TDMA for GSM takes the digital bit stream carried in the 200Khz radio channel and divides it up with a repeating pattern of frames. At the simplest level, the pattern repeats every eight frames. This gives eight time slots and, again with some simplification, this means that eight mobile phones can have a call connected at any one time. Each phone transmits and receives for only one-eighth of the time.

Each voice channel in GSM is therefore a digital bit stream at about one-eighth of the raw bit-rate supported by the 200Khz radio channel. The "about" has to be there because of overheads -- as elsewhere in life and engineering, there are always overheads. So if you need to transmit data instead of a digitized voice signal then the channel is there and the GSM standards engineers allowed for a whole variety of data "bearers" to use this channel. They called it the traffic channel (TCH). These bearers include various oddities like "speech followed by data" and the most widely used data service, non-transparent data at 9.6k bits per second (bps). This 9.6kbps service is what is used to connect laptop computers and WAP phones to the services they use. It provides error correction and flow control, which is what the "non-transparent" means in this context. It's usually referred to in GSM as CSD, circuit switched data.

Usually, network operators support CSD service through a modem interworking function. This means that a mobile makes a data call and the network routes the call to the modem interworking function, which then dials the number that the mobile supplied. This is very different from voice calls, where the network will route the call itself, often to another mobile on the same network. Mobile networks don't attempt to route data calls; they dial the requested number on behalf of the mobile and leave the routing to the external wire-line telephone network. The reason for this is that the network has no way of knowing what you want to do with a data call. Maybe you are contacting your ISP to pickup e-mail, maybe you are dialing your corporate network to set up a secure conversation to update the sales database or maybe the mobile is a burglar alarm contacting a security company. The network also does not know what speeds and compression are supported by the connection it dials. It has to take a "lowest common denominator" approach, just like your wire-line modem does when it dials your ISP. Like your wire line modem, it has to conduct a complex conversation modem-to-modem, to configure speeds and compression, every time you dial. This modem-to-modem conversation accounts for most of the delay when you set up a call from your laptop or WAP phone.

Some networks have their own ISP services, or can use ISDN to set up calls from the interworking function. In this case, there is no need for the long CSD call set-up. What the network cannot do is let you use the precious radio bandwidth only when you are transmitting or receiving data. Instead, as we have seen, you are occupying one TCH all the time while connected, regardless if you are using it to send data or not. This is not the most efficient way to utilize the network resources.

Most data communications is bursty and it would allow for a much more efficient use of the scarce radio channels if the mobile would only occupy resources when actually sending or receiving data. The mobile would then not imply a "cost" to the network operator when staying connected, which allows for the mobile to stay connected all day without the user paying for it. If you can make "staying connected" cheap then there is no need to dial up and connect calls, with all the waiting for modems that this usually implies.

GPRS:

The answer to bursty data traffic and high network utilization is packet switched networks. By using a packet switched technique there are no resources tied up while they are not in use. GPRS (General Packet Radio Service) introduces packet switching all the way from an origin server in an external network, to a mobile device. As with GSM, GPRS is standardized by ETSI and will initially support the IPv4 and X.25 protocols.

GPRS integrates with existing GSM systems and reuses the same radio network and the same transmission links between network nodes, for example between the Base Transceiver Stations (BTSs) and Base Station Controllers (BSCs) (which are the nodes that are responsible for the "last mile" out to the mobile device). The scarce radio resources are utilized much more efficiently by only using the GSM timeslots when required. Figure 1 shows how one of the eight time slots on single GSM radio bearer is monopolized by a circuit switched connection, such as a voice call.

In contrast, GPRS packets only use a timeslot when they need to. The diagram reads like a TV screen scan, left to right for the eight GSM timeslots, then down one row to the next repetition of the same eight. The economies of packet transmission should be fairly obvious. GPRS conversations can also use more than one timeslot when required delivering, within limits, bandwidth-on-demand.

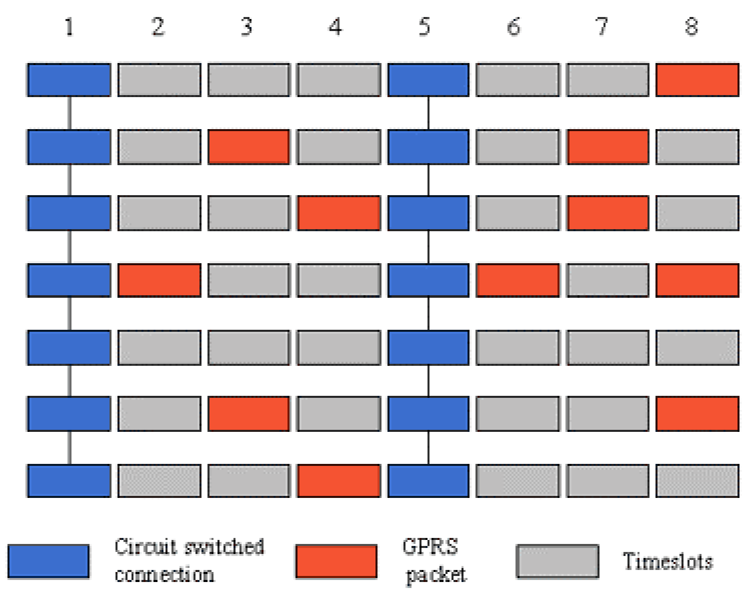

Figure 1: Differences in utilization of radio resources - GPRS and circuit switched mode.

So, GRPS can use one to eight timeslots for uplink and downlink traffic. Depending on the coding scheme, each timeslot can carry a different amount of data. There are four different coding schemes defined with different capacities. The capacity range from 9.05 kbps to 21.4 kbps where the slower ones provide varying amounts of error correction. For the network operators, GPRS means that they can quickly roll out packet switched services on their existing GSM network. It promises to let them maximize the use of their network resources, such as their scarce radio resources, in a flexible and dynamic way. It also makes it possible for an operator to use new billing models for data traffic, such as only charging for data transmitted not for the duration of the connection to the network.

However, some network operators may not allocate permanent time-slots for GPRS users only. Instead, they will use the dynamic option of GPRS, which means that GSM voice traffic is prioritized and packet switched data traffic is only transmitted on timeslots not currently in use. By doing this the operators can sell the capacity which is now wasted. Other operators will allocate channel capacity specifically for GPRS. The AT&T Wireless GSM/GPRS network will be optimized in this manner. Dedicated channels will be allocated specifically for GPRS data users.

For the end-users, packet switched data means that they can afford to be connected to the Internet or a company intranet "all the time." GPRS gives short set up times for higher layer protocols, which will make services much easier to reach, and higher bandwidth which means practicability of wholly new services.

The Network

GPRS is relatively easy to deploy for network operators operating existing GSM networks, since most of the upgrades are software upgrades in existing GSM nodes. On the hardware side there are two major new elements, GPRS Support Nodes (GSN), that have to be added. There are two types of GSNs. Both act like extended IP routers/gateways. They are called the Serving GPRS Support Node (SGSN), serving the mobiles themselves (called mobile stations, MS) and the Gateway GPRS Support Node (GGSN), which provides interworking with external packet data networks. These two different types of GSNs can be co-located but GPRS standards envisage that they are connected through an IP backbone network.

There are also other elements that have to be deployed for a commercial network, such as access servers and firewalls, but in this section we will concentrate on the GPRS specific elements.

GSN -- GPRS Support Nodes

To make the GPRS system work, we need the two different types of GPRS support nodes. We need SGSNs, which keep track of the mobile devices, both their location and their protocol state, so that we know where to send data and how. We need GGSNs, which connect a GPRS network external networks, such as Internet ISPs. SGSNs with other GPRS networks directly via border gateways.

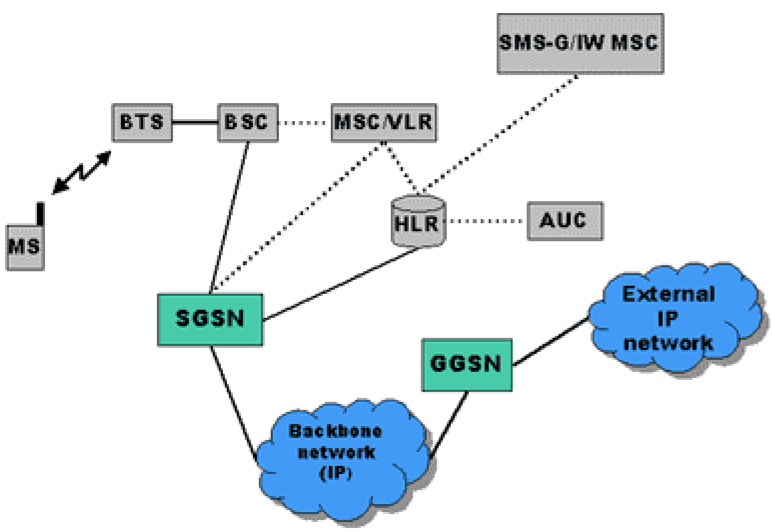

Figure 2: GPRS basic network structure

SGSN

Figure 2 shows how a SGSN logically fits into the GSM network. The SGSN is responsible for handling data packet traffic in a geographical area. It serves all mobiles (MSs) that are both attached to the GPRS service and located within the SGSN's service area. The SGSN provides authentication, mobility management, logical link management towards the MS, packet routing and transfer, plus connections to the Home Location

Register (HLR), and the GPRS packet control unit (PCU). The PCU takes care of the radio-specific parts of the GPRS protocol stack. Architecturally the PCU is part of the Base Station Subsystem (BSS) but it can be sited with a BSC or a BTS. Other connections, such as SGSN to MSC, are optional.

So, what does the SGSN do? Primarily, it tracks the movement of the subscriber to know where to send data packets providing a secure connection to the MS and that it routes traffic to and from the MS.

GGSN

The GGSN is the interface towards external IP networks and X.25 networks, as seen in Figure 2. The GGSN is acting as an access server and is responsible for routing incoming traffic to the correct SGSN (the traffic is carried using IP tunneling), i.e., it is responsible for setting up a logical link to the MS, through the SGSN. The GGSN also translates between data formats and translates signaling protocols and address information to enable communication between the different networks.

However, from an external IP networks point of view a GPRS network is just another sub-network, with the GGSN being a host owning all IP addresses of all subscribers served by that GPRS network. IP addresses can be assigned either statically or dynamically (see IP addresses section below).

The Devices

GPRS devices will, of course, come in all kinds of shapes and with different kinds of functionality. However, all GPRS devices can be divided into three different categories:

- **Class A** devices can operate GPRS simultaneously with other GSM services (such as a normal voice call).
- **Class B** devices can operate both GPRS and GSM services but not at the same time. The device must shift between the two modes but can be registered in the network for both.
- **Class C** devices operate exclusively GPRS services (although some devices may be manually set to handle GPRS or GSM, these are then seen as two different devices by the network, depending on the manual mode setting).

The first devices that will reach the market will be class B and class C devices. Class B devices can be attached to both GPRS and other GSM services, but can only operate one service at a time. If the device is attached to both GSM and GPRS services, but still remains in idle mode, it can monitor the paging channels for both circuit-switched and packet-switched services. However, whether or not the network can perform packet-switched paging is dependent on the network mode of operation.

When a device receives an incoming circuit-mode GSM paging request, it may be in GPRS mode. It can then suspend the GPRS activity and enter so-called dedicated mode and take the call. The GPRS mode is resumed when the device is returned to idle mode. As we will see later this behavior puts some demands on applications to run over a GPRS network.

Devices are being announced with different configurations of timeslots. Even with the slower of the coding schemes, this means that the delivered bit-rate can be up to about 40 Kbps downlink and 10 Kbps uplink.

As we have seen, the numbers of timeslots that can be used are up to the network operators. Networks may initially support only two or three downlink timeslots. This need not be a problem for services like current micro-browsers. I-Mode has shown that 9.6 Kbps per cell is sufficient to deliver a very attractive service provided packet switching shares the bandwidth efficiently.

Summary

GSM circuit switched data (CSD) access is slow, inefficient and expensive. GPRS will allow us all to "stay connected" cost-effectively by only using GSM timeslots when there are data to send. GPRS adds two major GPRS Service Nodes (GSNs) to the GSM architecture, SGSNs to service the mobiles and GGSNs to gateway into the external network world. Early GPRS devices will typically be able to support normal GSM voice and data but not simultaneously.

IP addresses (NEI – Network Entity Identifier)

The IP address (or NEI) is an identifier for a computer or device on a TCP/IP network. Networks using the TCP/IP protocol route messages based on the IP address of the destination. The format of an IP address is a 32-bit numeric address written as four numbers separated by periods. Each number can be zero to 255. For example, 1.160.10.240 could be an IP address.

Within an isolated network, you can assign IP addresses at random as long as each one is unique. However, connecting a private network to the Internet requires using registered IP addresses (called Internet addresses) to avoid duplicates. AT&T Wireless owns a pool of IP addresses for assignment to the various wireless devices connecting to its network.

The four numbers in an IP address are used in different ways to identify a particular network and a host on that network. The InterNIC Registration Service assigns Internet addresses from the following three classes.

- Class A - supports 16 million hosts on each of 127 networks
- Class B - supports 65,000 hosts on each of 16,000 networks
- Class C - supports 254 hosts on each of 2 million networks

The number of unassigned Internet addresses is running out, so a new classless scheme called CIDR is gradually replacing the system based on classes A, B, and C and is tied to adoption of IPv6.

CDPD and static IP's

Today all CDPD devices have static IP addresses assigned. Application servers never see the IP address of the AT&T Digital PocketNet device. Instead the IP address of the AT&T Digital PocketNet service gateway is exposed to the server. Applications use the subscriber ID to uniquely identify the PocketNet user. However, today's applications servers do have access to the IP address of non-PocketNet devices such as laptops, and PDA's for example.

GPRS and dynamic IP's

Wireless devices running on our GPRS network will take advantage of dynamic allocation of IP addresses when connecting (attaching) to the network. What this means to a wireless application developer is that if you have been addressing wireless devices by the NEI (or IP), you may need to change your application to handle the dynamic nature of the wireless device IP address. In some special or unique circumstances, some devices may be allocated a static address. But again, reliance on an IP address being unique to a device and always static may not be valid and should be taken into account when developing your application.

Be aware that devices such as AT&T Digital PocketNet & Pocket PC may be connected to a gateway within the AT&T Wireless GPRS network. As is the case today, the application server will not have access to the IP address connected to those gateways. In that case, a unique identifier will be available for applications that have a need to uniquely identify the subscriber. Today, the unique identifier for AT&T Digital PocketNet is the subscriber ID.

Markup Language Examples

The following code examples provide a comparison between CDPD applications written in HDML 3.0 and WML 1.1. An attempt has been made to make the user experience the same for both applications thereby demonstrating what is required for a ‘typical’ application rewrite from HDML to WML. In both examples Allaire’s Cold Fusion version 4.5 was used on the application server for database access. Cold Fusion tags have been left in the code for demonstration purposes. Alternative server side tools such as CGI scripting, Active Server Pages (ASP), or PERL for example may also be used. Wireless applications utilizing dynamic output rather than static output are more ‘typical’ of real-world applications on the AT&T Wireless network.

Note that some of the coding techniques and functions used are for demonstration purposes only. WML is loosely based on HDML as well as XML. Most (if not all) vendor specific extensions to the WML language standards may not be available on all gateways or browsers. Consult the WAP standard specifications for details on WML standard language specifications. Some spacing and indentation in the code examples has been eliminated for readability purposes.

HDML sample application

The first Deck (**DDP1.HDML**) displays an image then creates the “Main Menu” or CHOICE screen for the user. This deck contains 6 cards: 2 choice cards, 2 entry cards, and 2 display cards. The entry cards at the end of the deck allow user input. When the selects the “accept” soft key, either the **DDP2H.CFM** and **DDP3H.CFM** decks are ‘called’ depending on the entry card.

```
<HDML VERSION=3.0 PUBLIC=TRUE MARKABLE=TRUE TTL=86400>
<------------------------- File name DDP1.HDML ---------------------------------->
<---------Written by AT&T Wireless, DDP support engineers ------------------->

<DISPLAY NAME=StartDDP>
        <ACTION TYPE=ACCEPT Label=Continue TASK=GO DEST=#Main>
        <ACTION TYPE=SOFT1 Label=Quit TASK=cancel>

        <IMG SRC="DDP.bmp" ALT="Please wait, Image loading...">
</DISPLAY>
<CHOICE name=Main Markable=true Title="DDP Main Menu" Key=chx Ikey=numx idefault=1>
        <ACTION TYPE=ACCEPT task=go dest=#DisplayChoice Label=Select>
        <ACTION TYPE=SOFT1  Task=return Label=Quit>
        <center>AT&T Wireless
        <CE Task=gosub dest=#ln Value="name"><img icon="head1">Name lookup
        <CE Task=gosub dest=#co Value="company"><img icon="martini">Company lookup
        <CE VALUE="Specialty"><img icon="bolt">Specialty lookup
        <CE VALUE="Level"><img icon="star2">Level lookup
        <CE Task=gosub dest=#about VALUE="About"><img icon="question1">About
</Choice>
<CHOICE name=about Markable=true Title="About DDP" key=nbr>
        <ACTION TYPE=ACCEPT Icon="Phone" label=Call task=Call number=$nbr>
        <ACTION TYPE=SOFT1  task=return Label=Main>
        <center>DDP Database<br>
        <center>Written by<br>
        <center>DDP Support<br>
        <Line>For support call
        <CE Value="800-552-3373">ANS (Data)
        <CE Value="800-888-7600">Cust Care (Voice)
</Choice>
<DISPLAY NAME=DisplayChoice>
        You choose option ($numx), $chx lookup.
        <br>
        <br>
        This option is not enabled yet.  Please try another option.
</DISPLAY>
<ENTRY name=ln KEY=LName FORMAT=A10a>
        <ACTION TYPE=ACCEPT FRIEND=TRUE TASK=go DEST=DDP2H.cfm?LN=$LName LABEL=Find>
        Enter last name:
</ENTRY>
<ENTRY name=co KEY=Co FORMAT=A20a>
        <ACTION TYPE=ACCEPT TASK=go DEST=DDP3H.cfm?Cmpy=$Co LABEL=Find>
        Enter company name:
</ENTRY>
</HDML>
```

The second deck (**DDP2H.CFM**) is 'called' from the **DDP1.HDML** deck. Note that this file has a file extension of CFM. This deck includes Cold Fusion tags to demonstrate the dynamic output of this wireless application. When the mobile device requests this deck, the Cold Fusion application processes the request using the parameters passed (See ENTRY name=ln card above) to this deck. It dynamically builds the HDML deck based on the results of the database query and that deck is then sent back to the requesting mobile device.

```
<CFCONTENT Type="text/x-hdml">
<HDML VERSION=3.0 MARKABLE=TRUE PUBLIC=TRUE TTL=86400>
<------------------------- File name DDP2H.CFM ------------------------------------->
<---------Written by AT&T Wireless, DDP support engineers ------------------->
<cfquery name="lastname" datasource="ddp">
	SELECT firstname, lastname, phone
	FROM developer
	where lastname='#url.LN#'
	order by Lastname, firstname;
</cfquery>
<CHOICE NAME="List" KEY=nbr>
<cfoutput>
	<cfif #lastname.RecordCount# EQ 0>
		<ACTION TYPE=ACCEPT task=return clear=true label=OK>
		No records found.<br>
		<CE value="try again">Select again
	<cfelseif #lastname.RecordCount# EQ 1>
		<ACTION TYPE=ACCEPT Icon="Phone" label=Call task=Call number=$nbr>
		<ACTION TYPE=SOFT1  task=return clear=true Label=Main>
		1 record found.<br>
	<cfelse>
		<ACTION TYPE=ACCEPT Icon="Phone" label=Call task=Call number=$nbr>
		<ACTION TYPE=SOFT1  task=return clear=true Label=Main>
		#lastname.RecordCount# records found.<br>
	</cfif>
</cfoutput>
<cfoutput query="lastname">
	<CE value=#phone#>#lastname#, #firstname#
</cfoutput>
</CHOICE>
</HDML>
```

The third and final deck (**DDP3H.CFM**) is also 'called' from the **DDP1.HDML** deck. This file also has a file extension of CFM. The deck is essentially the same as **DDP2H.CFM** except for the output.

```
<CFCONTENT Type="text/x-hdml">
<HDML VERSION=3.0 MARKABLE=TRUE PUBLIC=TRUE TTL=86400>
<------------------------- File name DDP3H.CFM ------------------------------------->
<---------Written by AT&T Wireless, DDP support engineers ------------------->
<cfquery name="company" datasource="ddp">
	SELECT firstname, lastname, phone
	FROM developer
	where Company='#url.Cmpy#'
	ORDER BY lastname, firstname;
</cfquery>
<CHOICE NAME="List" KEY=nbr>
<cfoutput>
	<cfif #company.RecordCount# EQ 0>
		<ACTION TYPE=ACCEPT task=return clear=true label=OK>
		No records found.<br>
		<CE value="try again">Select again
	<cfelseif #company.RecordCount# EQ 1>
		<ACTION TYPE=ACCEPT Icon="Phone" label=Call task=Call number=$nbr>
		<ACTION TYPE=SOFT1  task=return clear=true Label=Main>
		1 record found.<br>
	<cfelse>
		<ACTION TYPE=ACCEPT Icon="Phone" label=Call task=Call number=$nbr>
		<ACTION TYPE=SOFT1  task=return clear=true Label=Main>
		#company.RecordCount# records found.<br>
	</cfif>
</cfoutput>
<cfoutput query="company">
	<CE value=#phone#>#lastname#, #firstname#
</cfoutput>
</CHOICE>
</HDML>
```

WML sample application

The first Deck (**DDP1.WML**) displays an image then creates the “Main Menu” or SELECT screen for the user. This deck contains 6 cards. Note the difference between HDML and WML. HDML has different *types* of cards while everything in WML is a card. The WML cards serve different functions based on what is coded within the card. The last two cards allow user input. When the selects the “accept” soft key either the **DDP2W.CFM** and **DDP3W.CFM** decks are ‘called’ depending on which card is active card. Note that method=post is being used which is the preferred way of returning data to your application.

```
<?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" forua="true"/>
</head>
<card id="main" title="DDP Main Menu">
        <do type="accept">
                <go href="#displaychoice"/>
        </do>
        <do type="options" label="Quit">
                <prev></prev>
        </do>
        <p align="center" mode="wrap">
                AT&amp;T Wireless
        </p>
        <p align="left" mode="nowrap">
        <select title="DDP Menu WML" name="chx" iname="numx" ivalue="1">
                <option value="name" onpick="#ln">Name lookup</option>
                <option value="company" onpick="#co">Company lookup</option>
                <option value="Specialty">Specialty lookup</option>
                <option value="Level">Level lookup</option>
                <option value="About" onpick="#about">About</option>
        </select>
        </p>
</card>
<card id="about" title="About DDP">
        <do type="accept" label="Call">
                <go href="wtai://wp/mc;$phnbr"/>
        </do>
        <p align="center" mode="wrap">
                DDP Database<br/>
                Written by<br/>
                Scott McEntire<br/>
        </p>
        <p align="left" mode="nowrap">
                For support call
        <select name="phnbr">
                <option value="8005523373">ANS (Data)</option>
                <option value="8008887600">Cust Care (Voice)</option>
        </select>
        </p>
</card>
<card id="displaychoice">
        <p>
```

```
                    You choose option ($numx), $chx lookup.
                    <br/>
                    <br/>
                    That option is not enabled yet. Please try another option.
          </p>
</card>
<card id="ln">
          <do type="accept" label="Find">
                    <go href="DDP2W.cfm" method="post">
                    <postfield name="ln" value="$(ln)"/>
                    </go>
          </do>
          <p>
          Enter Last Name:
          <input name="ln" title="Input name" type="text" format="A10a" emptyok="false"/>
          </p>
</card>
<card id="co">
          <do type="accept" label="Find">
                    <go href="DDP3W.cfm" method="post">
                    <postfield name="co" value="$(co)"/>
                    </go>
          </do>
          <p>
                    Enter Company Name:
                    <input name="co" title="Input company" type="text" format="A20a" emptyok="false"/>
          </p>
</card>
</wml>
```

The second deck (**DDP2W.CFM**) is ‘called’ from the **DDP1.WML** deck. Note that this file has a file extension of CFM. This deck includes Cold Fusion tags to demonstrate the dynamic output of this wireless application. When the mobile device requests this deck, the Cold Fusion application processes the request using the parameters ‘posted’ (See card id=”ln” above) to this deck. It dynamically builds the WML deck based on the results of the database query and that deck is then sent back to the requesting mobile device.

```
<CFCONTENT Type="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>
<head>
   <meta http-equiv="Cache-Control" content="max-age=0" forua="true"/>
</head>

<cfquery name="lastname" datasource="ddp">
        SELECT firstname, lastname, phone
        FROM developer
        where lastname='#ln#'
        order by Lastname, firstname;
</cfquery>

<card id="main" title="Last name lookup">
<cfif #lastname.RecordCount# EQ 0>
        <cfoutput>
                <p>No Records found. Press OK to try again.</p>
        </cfoutput>
<cfelse>
        <cfoutput>
                <do type="accept" label="Call">
                        <go href="wtai://wp/mc;$phnbr"/>
                </do>
                <do type="options" label="back">
                        <prev></prev>
                </do>
                <p>#lastname.RecordCount# records found<br/></p>
                <p><select name="phnbr">
        </cfoutput>
        <cfoutput query="lastname">
                <option value="#phone#">#lastname#, #firstname#</option>
        </cfoutput>
        <cfoutput>
                </select>
                </p>
        </cfoutput>
</cfif>
</card>
</wml>
```

The third and final deck (**DDP3W.CFM**) is also 'called' from the **DDP1.WML** deck. This file also has a file extension of CFM. The deck is essentially the same as **DDP2W.CFM** except for the output.

```
<CFCONTENT Type="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>
<head>
   <meta http-equiv="Cache-Control" content="max-age=0" forua="true"/>
</head>
<cfquery name="company" datasource="ddp">
        SELECT company, firstname, lastname, phone
        FROM developer
        where company='#co#'
        order by lastname, firstname;
</cfquery>

<card id="main" title="Company name lookup">
<cfif #company.RecordCount# EQ 0>
        <cfoutput>
                <p>No Records found. Press OK to try again.</p>
        </cfoutput>
<cfelse>
        <cfoutput>
                <do type="accept" label="Call">
                        <go href="wtai://wp/mc;$phnbr"/>
                </do>
                <do type="options" label="back">
                        <prev></prev>
                </do>
                <p>#company.RecordCount# records found<br/></p>
                <p><select name="phnbr">
        </cfoutput>
        <cfoutput query="company">
                <option value="#phone#">#lastname#, #firstname#</option>
        </cfoutput>
        <cfoutput>
                </select>
                </p>
        </cfoutput>
</cfif>
</card>
</wml>
```

Frequently Asked Questions (FAQ's)

Will my CDPD device work on the GPRS network?

No. CDPD and GPRS devices use different modems with incompatible transmission standards.

Will my HDML application run on a GPRS network?

It depends. AT&T Digital PocketNet service applications will initially continue to pass through our Openwave (Phone.com) gateway. Backwards support for HDML applications over GPRS networks will be provided for devices with native HDML browsers. All of the GPRS PocketNet devices that AT&T Wireless will support will run WML natively on the device. Any HDML content accessed by a GPRS device (native WML) will be translated into WML at the gateway. However, some of the new GSM/GPRS devices will not have an Openwave browser and will not support HDML or any of the OpenWave extensions. Translation of HDML to WML at the gateway *may* allow these HDML applications to function but the functionality of the application may be severely crippled. HDML tags used in the application that have a one-for-one translation to WAP standard WML will translate at the gateway but may behave differently on the various WAP browsers. AT&T Wireless has developed a HDML style guide that addresses this issue in more detail.

What version of WML is supported with GPRS?

The version of WML depends on the version of the UP.link gateway and the version of the browser in the wireless device. At the date of this writing the AT&T Wireless UP.link version is 3.2.3.4. That will soon be upgraded to version 4.x. WML 1.1 content is translated into HDML at the gateway for handsets that have phone.com browsers below version 4.x. Depending on the browser vendor and version on the handset, WML or HDML will be sent to the phone.

When will GSM/GPRS be available in my area?

AT&T Wireless GSM/GPRS network commercial availability will be available in selected markets in the continental United States in 2001. Specific market availability dates have not been made public at this time.

Will the current TDMA and CDPD networks continue to be supported?

Yes. We are fully committed to maintaining our current network operations while at the same time deploying the next generation of wireless networks. This gives our customers a smooth migration path from the current wireless technologies to the newer technologies as they become available.

Helpful Web sites

Here are a limited number of sites that provide helpful information for wireless application development.

- http://www.att.com/wirelessir/
  AT&T Wireless Investor Relations site.
- http://www.attws.com/business/lcorp/explore/wireless_ip/developers/
  AT&T Wireless Data Developer Program (DDP)
- http://www.attws.com/business/lcorp/explore/wireless_ip/bizopps/
  Business opportunities/Solutions. Data Solutions Providers (DSP) and Value Added Resellers (VAR) program.
- http://www.mobilegprs.com/
  Good information on GPRS from a European (GSM – circuit switched) perspective.
- http://www.wapforum.com
  WAP forum site. View and download various WAP specification documents.
- http://developer.openwave.com/resources/uiguide.html
  Openwave’s Top 10 Usability Guidelines for WAP Applications
- http://developer.openwave.com/technotes/hdml2wml/index.html
  Migrating from HDML to WML. Implementation Guidelines for Web developers
- http://www.ericsson.com/developerszone/
  Ericsson Developers' Zone
- http://www.mitsubishiwireless.com/manuals/
  Manuals for Mitsubishi phones including the T250
- http://www.motorola.com/developers/wireless/
  Motorola developer’s site

Definitions and Terms

The following list is not an all-inclusive list of acronyms and terms.

2.5G	**Second Generation Wireless** included digital phones. **2 ½ Generation Wireless** (or 2.5G) includes digital wireless voice and high-speed data.
3G	**Third Generation Wireless.** Digital voice plus high-speed data and global roaming. Known as IMT 2000 by the ITU and implemented in Europe as UMTS and CDMA 2000 in North America. Goals are high quality multimedia and advanced global roaming.
Card	A single WML unit of navigation and user interface. May contain information to present to the user, instructions for gathering user input, etc. HDML also uses cards.
CDPD	Cellular Digital Packet Data.
Deck	A collection of WML cards. A WML deck is also an XML document. HDML also uses decks.
EDGE	**Enhanced Data rates for Global Evolution.** Increased data throughput in GSM and TDMA systems to 384 Kbps.
GGSN	**Gateway GPRS Support Node.**
GPRS	**General Packet Radio Service.** A Packet radio-access technique based on GSM radio to transfer data in an efficient manner optimizing the use of network resources. It re-uses existing GSM radio technology.
GSM	**Global System for Mobile communications.** A technical standard for global mobile networks.
GSN	**GPRS Support Node.** Includes the **SGSN** and **GGSN**.
HDML	Handheld Device Markup Language. Forerunner of WML.
Phone.com	Now known as **Openwave Systems Inc**. and previously known as Unwired Planet. Originally created HDML. Created handset browsers (UP.browser) and gateway server (UP.link) that serve as proxy agent for wireless devices. See http://www.openwave.com/index.html for more information.
SGSN	**Serving GPRS Support Node.**
UMTS	**Universal Mobile Telecommunications System.** A new generation technology for rapidly moving data and multimedia over wireless devices. The European implementation of the 3G wireless phone system. UMTS provides service in the 2GHz band and offers global roaming and personalized features. Designed as an evolutionary system for GSM network operators, multimedia data rates up to 2 Mbps are expected.
WAP	**Wireless Application Protocol.** WAP is an open, global specification that empowers mobile users with wireless devices to easily access and interact with information and services instantly. See the WAP forum Web site (http://www1.wapforum.org/) for more information.
WML	**Wireless Markup Language.** WML is a markup language based on XML and is intended for use in specifying content and user interface for narrowband devices, including cellular phones, PDA's, and pagers. WML is the markup language for WAP.
WMLScript	A scripting language used to program the mobile device. WMLScript is an extended subset of the JavaScript scripting language.
XML	**Extensible Markup Language** is a World Wide Web Consortium (W3C) standard for Internet markup languages, of which WML is one such language. XML is a restricted subset of SGML.