

VIDEOAPPLICATIONS

*PIM*mobile



DEVELOPER GUIDE

Cestel 

INTRODUCTION

This document is the development guide for 3G phones multimedia applications, based on Cestel's PIMobile environment (PIMobile from now on).

PIMOBILE GENERAL WORKING OURLINE

PIMobile is a service content platform, based on Cestel Technology. The objective of PIMobile is that calls routed to him by DNIS, ANIS, transfer from other video application or any other criteria to be relayed the image and audio of a web browser, this browser has a navigation by URL of Internet Connected web services.

PIMobile application have a associated URL as starting point, through this URL PIMobile access the content and relays to the mobile phone the image and sound that the web browser would show. This starting URL is configured through its video mobile content platform's administrative tool.

From the moment the starting URL image is shown, PIMobile manages the conversion of pulsed keys on the phone (DTMF digits) to be retrieved and shown on the 3G phone. Meaning, the pulsed digits becomes the equivalent to the clicks on the links on each retrieved web page shown on the phone.

For this task PIMobile has an Internet Explorer web browser for each 3G phone it answers at the same time, thus at any given moment holds an array of browsers to which allocates and ends sessions generated by each call.

Therefore, PIMobile is capable of performing its function as long as the websites that are accessed are ready to work under Internet Explorer, and if used on the pages accessed any additional object or embedded in a web page (eg a display of videos or similar), it must be previously registered with the server that hosts the software PIMobile.

From the point of view of the application developer, the execution of a PIMobile scheme based application therefore is as follows

Therefore an application PIMobile lies not actually in the services platform or in the space or environment PIMobile, but is actually a Web application that resides on a conventional web server (or web servers linked to one another at logical level by links).

Because the final destination of the application PIMobile is not a web browser, but a mobile phone, web applications must meet certain requirements set out in the next chapter, but only in the format shown and to the commands capture not in terms of the development tools and technologies themselves.

That means an application PIMobile can be develop with any tool or usual procedure in web development: JSP, PHP, ASP, ASPX, FLASH, Javascript ... etc.

Any programmer therefore trained in any of these tools is trained too on the development for PIMobile, just by following and taking into consideration the issues listed and explained in this manual.

Therefore in the rest of this manual is assumed that the reader is familiar with the concepts and knowledge of general use in web applications development: URLs, links, frames, HTML elements, JavaScript etc.

PIMOBILE WEB APPLICATION REQUIREMENTS

Any Web application is likely to run in the environment PIMobile if it meets the requirements listed below.

What does "likely to run" in PIMobile mean? It means that it will be properly displayed / heard in a 3G mobile phone and that simply by pressing the DTMF phone keys it will navigate (ie, link URLs).

Therefore this is equivalent to saying that a Web application is "certified PIMobile."

The main requirements are as follows, and each of them will make some comments and suggestions to facilitate the conversion of existing web applications with minimal effort.

- Screen Size
- Managing keyboard
- Internet Explorer 6 (IE6) Compatible
- Implementation in a single window or iframe

SCREEN SIZE

The screen of a cell phone is about the size of standard QCIF that is 176 pixels wide by 144 high (greater width than height, although in some phones might seem that the vertical dimension is greater)

Therefore, if the web application employs more space on the screen, only 176x144 pixels from the upper left corner of the information that would show an internet explorer browser is shown.

For example, if the entry point of our application PIMobile was

[Http://www.cestel.es](http://www.cestel.es)

The image displayed in a browser Internet Explorer 6 (IE6) for the URL is as follows



PIMobile Hill only send to the phone and image of 176x144 belonging to the upper left corner, like this one:



However, if necessary and the web application accessed is prepared, PIMobile has the full picture and through JavaScript a scroll could be done to show areas initially hidden and out of the window of 176x144 displayed.

In other words, we could display on the mobile for example the red framed area in the picture below. How to do this is addressed below. Applications for scrolling viewing area would be for example

- Move by the image of a map
- Vertical scrolling on a "long" Web screen, such as an electronic publication of a newspaper.



In any case it would be normal to display on your phone a website QCIF size, as in this example menu



Or a video image or picture adapted to that size as in these examples of video and webcam



Webcam



Video de un Telediario

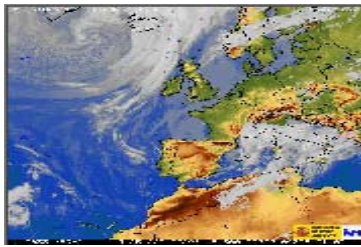


Imagen de satélite

A good practice could be including in the HTML of the developing pages of a PIMobile a DIV element QCIF size. This way during development if pages are invoked from a PC we will only see the part that would show on a phone. If we design a page not adequate to QCIF size we will notice simply by loading it in a conventional PC.

Consider the following example:

The DIV element through CSS has been given a frame to visualize its limit. The style applied to DIV can be included in a file CSS, to reuse in the different page of the PIMobile Website. If we invoke this page:

<http://www.pimobile.es/samples/demodiv1.htm>

from a PC browser we will see on screen

The black frame of the DIV marks the equivalent space of the mobile screen. In this case the mobile will show the complete content.

However if we enlarge the width of the table and insert more text, for example changing the table code to

The result, shown in the image: The whole text is not in the frame anymore (there is even scroll bars) and the result will be then inadequate for a mobile phone.

The web developer for PIMobile therefore should keep in mind that the space available through normal operation (with no zoom and scroll, but as we say it is feasible) is reduced and therefore

- The text to be included must be limited in size
- Images / videos must have enough quality in QCIF size

Thus at the time of adapt or develop a PIMobile Web application will be necessary to clearly define a strategy and usability to make viable and flexible visual presentation on the screen.

MANAGING KEYBOARD

In a conventional web application (for a PC browser with minimum resolution of 640x480), once the browser displays a page it is possible to click on the links associated with texts or images, which causes the browser to ask the URL assigned to the link clicked, and display a new web page as result.

This is the usual way to work, but applications PIMobile, by the very nature of the 3G mobile applications is not possible but to send strokes of DTMF digits (numbers 0 to 9, key * # key).

Simulating, like when we type the text of an SMS message, we can also send text and special characters: eg three quick beats of the key 2 are equivalent to the letter C. In any case, those are keystrokes on the phone's DTMF keypad.



But from a conventional web application it's possible to activate links to a page by pressing numeric digits or the * key or #. This will need to minimally alter the page, adding Javascript code identifying each link to consider and assigning it a key.

For example, it is possible to make the following link (shown HTML code) to be activated in a web application conventional, plus PIMobile, by pressing the keyboard.

[Go tp PIMobile Web](#)

For those there basically two methods:

- the first to be called "direct management" allowing through JavaScript to directly receive the code of the key pressed and act accordingly: for example to activate a link on the page, or scrolling, zooming or anything else achievable via javascript . It is appropriate if you want complex effects, such as zooming, scrolling, resizing images, active or deactivate layers ...
- The second, called "management by links" simply assigning, with very little code, numerical options to links on the website. It is the most appropriate for menus or web pages containing multiple links and whose mission is to serve primarily as an intermediate step to another.

As "evolution" of the direct management, it's also possible to manage the keyboard to enter data into a form. For this PIMobile provides a Javascript library, which will be discussed in the chapter [Forms](#).

DIRECT KEYBOARD MANAGEMENT.

1º) Add on the page, in the element <BODY> the attribute

```
onkeypress = "keyAction ()"
```

This means that before any keyboard pulsation (numerical or not) browser calls the function indicated, without further action in this regard.

2º) Define the previous function, which will answer the keystrokes. Within this function is possible to retrieve the key pressed, which will be coded on the basis of the table attached below. While in a conventional web application this function would receive pulsations of any key in a PIMobile only receives the DTMF digits, reason we shall confine ourselves to consider only these keys on the function.

Keycode	Dígito DTMF
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
42	*
35	#

Example of a function to attend the keyboard

```

<script>
function keyAction() {
    var key = String.fromCharCode(window.event.keyCode)
    switch( key ) {
        case '0':          // code to attend key 0
                        // activate the first link of the page.
                        document.links[0].click()

break

        case '1':          // code to attend key 1
                        // Navigate to a URL
                        location = 'http://www.cestel.es?index=1'
                        break
        case '2': // code to attend key 2
                        // resize a Picture.
                        document.images[0].width=176
                        break
        case '*': // code to attend key *
                        // go to previous page
                        history.back()
                        break
        default: // no answer to other keys.
                        break
    }
    // to finish nullify the event
    window.event.keycode = null
}
</script>

```

KEYBOARD MANAGEMENT BY LINKS

1º) Include the javascript code of the keyboard management “3G Navigation Handler” not forgetting the header of the page.

```

<script
src="http://demo3g2007.cestel.es/programsQCIF/lib/cestel.3g.NavHandler.js"></script>

```

(The example URL points to CESTEL web server, also you can put the file with the rest of the web application files, to make your own adjustments.)

2º) define the keys assignments through the javascript code as in the following example:

```

<html>
<head>
<script
src="http://demo3g2007.cestel.es/programsQCIF/lib/cestel.3g.NavHandler.js">
</script>
</head>

<body topmargin="0" leftmargin="0">
<table width="176">
<tr>

```

```

<td align="center" width="176">
  <br>
  Elija idioma / Choose language<br>
  <br>
  <br><br>

<br>
  <a href="menu1.htm" id="A1">Menu 1</a>
  <a href="menu2.htm" id="A2">Menu 2</a>
  <a href="menu3.htm" id="A3">Menu 3</a>
  <a href="menu4.htm" id="AA">Exit</a>
</td>
</tr>
</table>

<script>
  var commands = new Array();
  // links association options.
  commands["1"]="A1";
  commands["2"]="A2";
  commands["3"]="A3";
  commands["*"]="AA";
  // Do not forget this line
  window.document.onkeypress = navigationHandler;
</script>
</body>
</html>

```

In the previous example the '1' key would navigate to the relative direction 'menu1.htm' of the actual URL, y la tecla "*" a 'menu4.htm'.

As seen the unique javascript code necessary is the one that defines the association key-link. The only other thing to take on account is to assign an identifier to each link (id="A3" in the example is the assignation of de identifier to the third link.

INTERNET EXPLORER 6 (IE6) COMPATIBLE

As explained in the general pattern of operation, applications PIMobile will run in the server PIMobile through an object Internet Explorer. Currently version is 6.0 by which any application PIMobile must display and function correctly in this version.

Simply access the application developed from a PC equipped with this version to certify that the operation is correct in PIMobile.

IMPLEMENTATION IN A SINGLE WINDOW OR IFRAME

Applications PIMobile are the relayed image of a single window of an Internet Explorer browser. Therefore alert messages (such as alerts Javascript) or pop-up windows are not displayed on the phone, so it should be avoided.

Especially avoid alerts of JavaScript, a user could not close those by any key on his phone, thereby the PIMobile application would be blocked. This kind of windows, occasionally used

for debugging should only be used in previous versions or purification applications and designed to run as conventional applications (in a PC).

ADVANCED PROGRAMMING OF PIMOBILE APPLICATIONS

In this chapter we explain how to get the PIMobile applications to do determined actions not usual on a conventional web application but that in a 3G environment are specially useful or needed.

INCLUSION OF OBJECTS

As in a conventional application, it is possible insert calls to objects through HTML tags of the form

```
<object classid="CLSID:xxx-xxx-....."  
  type="application/x-oleobject" id="picture"  
  width="176" height="133"  
  codebase="....." >  
  <param name="..." value="...">  
  <param name="..." value="...">  
</object>
```

It should be borne in mind that the area allocated to the object must be smaller than it would be normal for a conventional application. This size of the area must be equal or less than the standard QCIF (except if made for scrolling). The size is controlled by attributes WIDTH and HEIGHT of the OBJECT tag.

In any event it is necessary to check how the resized images is when shown by the object, it might present stripes filling in the areas top, bottom and / or sides, or not even admit to be displayed with the size adapted to our needs.

Another important issue is that the object must be registered on the server PIMobile if it is to be loaded on the Internet Explorer browser and display correctly on the phone. Therefore, no objects that may be used in conventional web applications requiring a previous download of something ActiveX, Java or similar can be used. Only may be used if previously requested to CESTEL to be registered on the object's PIMobile server.

The two most common types of object (Windows Media and Flash) are already installed on the server PIMobile, so those can be used directly, and run on PIMobile in the same way as they would on a conventional web application.

Both types of object (and others which could be used) must pay attention to:

- Enable auto start mode of reproduction, if it is a video, as a general rule will be difficult to convert the pressing a key on the phone to a click on a particular area of the object that spawns reproduction
- Off-display control panel which often carry these objects (buttons play, stop, pause, etc.) because it will also be difficult to emulate the clicks in a particular position of the control panel to get the boot, stop, etc.

However, objects tend to be controllable via JavaScript (by events). In section direct management of the keyboard we have seen that it is possible to translate a pulsation on the keyboard to Javascript code, and that code could be valid to stop, start and so the reproduction of videos in an object of this type.

Next text discusses two examples of objects including Windows Media and Flash, but not being this comprehensive guide of the parameters and scope of these objects.

For that check out:

Flash:

http://kb.adobe.com/selfservice/viewContent.do?externalId=tn_12701

Windows Media:

<http://msdn2.microsoft.com/en-us/library/ms983653.aspx>

FLASH OBJECT INCLUSION

```
<object id="Player" width="176" height="194" valign="top"
  classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000">
  <param name="allowScriptAccess" value="sameDomain" />
  <param name="menu" value="false" />
  <param value="/comunes/multimedia/video/1_a/video_1_a.swf?update=200704261431"
  name="movie" />
  <param
  value="url_fv=%2Fcomunes%2Fplayer%2Fmm.php%3Fxfref%3D20070524elpepunac_5.Ves
  &referrer_fv=http%3A/www.elpais.com/videos/espana/Acebes/acusa/Zapatero/reabrir/herid
  as/pasado/generar/division/elavid/20070524elpepunac_5/Ves/&memory_fv=0&autoplay_fv=
  1&id_fv=842983&pagead_fv=" name="FlashVars" />
  <param value="best" name="quality" />
  <param value="opaque" name="wmode" />
  <param value="#ffffff" name="bgcolor" />
</object>
```

WINDOWS MEDIA VIDEO OBJECT

Fundamental parameters are highlighted.

- Autostart = so the reproduction will Start by itself.
- Rate = the speed of playback
- Uimode = This is the value that indicates to show the control panel
- URL = URL of the video source (formats http, mms, rtsp...)

```
<object classid="CLSID:6BF52A52-394A-11d3-B153-00C04F79FAA6"
  type="application/x-oleobject"
  id="Picture"
```

```

width="176" height="133"

standby="Loading content..."
codebase="http://activex.microsoft.com/activex/controls/mplayer/en/nsmp2inf.cab#Version=
5,1,52,701">

<param name="autoStart" value="true"/>

<param name="balance" value="0"/>

<param name="currentPosition" value="0"/>

<param name="enableContextMenu" value="false"/>

<param name="enabled" value="false"/>

<param name="fullScreen" value="false"/>

<param name="mute" value="false"/>

<param name="playCount" value="1"/>

<param name="rate" value="0.5"/>

<param name="uiMode" value="none"/>

<PARAM NAME="windowlessVideo" VALUE="true">

<PARAM NAME="URL"
VALUE="mms://stream.farreachtech.com/newsmyrnabeachcam">

<param name="volume" value="70"/>

</object>

```

FORMS

In conventional web applications based on HTML is frequent the use of forms, defined by tags FORM, INPUT and SELECT.

To complete text fields is people use of alphabetical keyboard on a PC, and to activate, deactivate or list camps radio, checkbox and select use the mouse and the arrow keys.

As discussed in chapter management keyboard, by pressing DTMF digits is possible to simulate the alpha and special characters (comma, point, space etc.), thus it will also be possible to complete the text fields in forms.

To that mean is provided a Javascript code file, which included in the code of the HTML page with the form to handle allows filling your data from the phone, and that each field and the form as a whole be validated.

A simple example is the following one:

<http://www.pimobile.es/samples/demo.htm>

```

<html>

<head>

  <script src="forms.js"></script>

  <!-- control the text aspect (size, color...) with CSS -->

  <style type="text/css">

    td {font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10px; color:
171717;}

    input {font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10px;
color: 171717;}

  </style>

</head>

<body topmargin="0" leftmargin="0" onload="focusField( 'fform', 'field1' )">

  <form name="fform" action="action.htm">

    <table>

      <tr>

        <td colspan=2 align=center>Sample form</td>

      </tr>

      <tr>

        <td>Field1</td>

        <td><input type=text name=field1 id=field1 size=4 maxlength=4
onkeypress="tecla(kLetAlfa,false)" tabindex=1></td>

      </tr>

      <tr>

        <td>Field2</td>

        <td><input type=text name=field2 id=field2 size=10 maxlength=10
onkeypress="tecla(kLetAlfaNum,false)"

```

```

tabindex=2></td>
    </tr>
    <tr>
        <td>Field3</td>
        <td><input type=text name=field3 id=field3 size=8 maxlength=8
                                onkeypress="tecla(kLetNum,true)"
tabindex=3>##=Submit</td>
    </tr>
    <tr>
        <td colspan=2>Fill every field, #=Next field</td>
    </tr>
</table>
</form>
</body>
</html>

```

Points to highlight from the code above, and which are necessary conditions for achieving a valid form for an application PIMobile:

- It includes a call to Javascript code from file forms.js. In this file resides the coded functions focusField and "tecla" (Spanish for key).
- The form includes only text input fields, but could also include hidden
- The TAG body includes a call for placing the focus on a particular field of the form, which might not be the first one. The parameters are
 - Function focusField (nform, nfield)
 - Nform = name of the form on which to set the focus, whether it is null or non-existent it will be the first on page.
 - Nfield = name of the field to focus in the form, whether it is null or non-existent will be the first one.
- Each input must contain an attribute that defines its name within the form (name) and its other identifier element in the DOM (Document Object Model) of the page (id).
- Each input must contain an attribute tabindex to define the sequence of "jump" between fields, it's to say, the first will have value tabindex = 1, the second tabindex = 2 etc.
- Each input must state the function "tecla" to be enforced in every keystroke in this field, ie the attribute onkeypress. The parameters are
 - function tecla (arrayValores, validateField, validateFunction)

Therefore an application PIMobile using Javascript can move the box to show other parts of the whole picture of the web page loaded in Internet Explorer.

This can be useful in certain applications that seek to show an image (still picture or video) than in QCIF size is not appreciated in sufficient detail, for example

- Webcams
- Maps
- Graphics
-

Another interesting possibility is to resize images (or objects) whose size is not adequate to QCIF, e.g. zoom in or out of them. Combined with the scroll this facilitates inspecting the detail of an image considerably bigger in size than QCIF, such as a fixed webcam or video camera

The basic example would be as the one below:

<http://www.pimobile.es/samples/demoscroll.htm>

It shows, updated in real time, the image of a still webcam (image taken from time to time) for a ski resort

```
<html>
  <head>
    <style type="text/css">
      td {font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10px; color:
#171717;}
      input {font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10px;
color: #171717;}
    </style>
  </head>

  <body onload="loaded()"
    onkeypress="keyStroke()" topmargin="0" leftmargin="0">

    <!-- help screen, command 0 hides/unhides -->

    <div width="100" height="80" id="help" style="font:10px/12px
Arial,Helvetica,sans-serif; border:solid 1px
#666666;
```

```

width:100px; height:80px; padding:1px; position:absolute; z-index:100;
visibility:visible; color:#333333;

top:15px; left:15px; background-color:#ffffcc; layer-background-
color:#ffffcc;">

0=help on/off<br>
9=exit<br>
7=zoom+<br>
3=zoom-<br>
2,8,6,4=move up/down/right/left

</div>

<!-- lower permanent options menu -->

<div width="176" height="8" id="help1" style="font:10px
Arial,Helvetica,sans-serif; border:solid
1px
#666666;

width:175px; height:15px; padding:1px; position:absolute; z-index:100;
visibility:visible; color:#333333;

top:129px; left:0px; background-color:#ffffcc; layer-background-
color:#ffffcc;">

<span id=txtcamara1><b>0 Help 7/3 Zoom+/- 2,8,6,4
Move</b><br></span>

</div>

<script src=scroll_zoom.js></script>

<script>

// start position ( coords of the point to be shown at screen center)

var x = 100

var y = 200

// URL of the image to show

```

```
var imageURL = 'http://www.valdesqui.es/cam/valdemartin.jpg'  
  
// URL to exit (command 9)  
  
var urlQuit = null  
  
document.write('<br><br><br>')  
  
var windowWidth = 176  
  
var windowHeight = 144  
  
var windowWidthHalf = parseInt(windowWidth/2)  
var windowHeightHalf = parseInt(windowHeight/2)  
  
var menuHeight = 15 // height of the permanent menu  
  
//real screen height, considering bottom menu  
var actualScreenHeight = windowHeight-menuHeight  
  
  
  
// pixels from top for permanente menu (width of menu=144-topMenu)  
var topMenu = windowHeight-menuHeight  
  
// pixels from left for permanente menu (width of menu=144-topMenu)  
var leftMenu = 0  
  
  
  
// max Zoom factor allowed  
  
var topZoom = 8  
  
// starting zoom level  
  
var zoom = 1  
  
// pixels for every zoom level increment  
  
var pixZoom = 80  
  
// pixels for every scroll incremente  
  
var pixScroll = 30  
  
  
  
// if true, the image if initially fitted to 176x144 ("full screen")
```

```
var startFit = true

// allow fit ( command 5 )

var allowFit = true

// allow move to home position (upper left 0,0) (command 1)

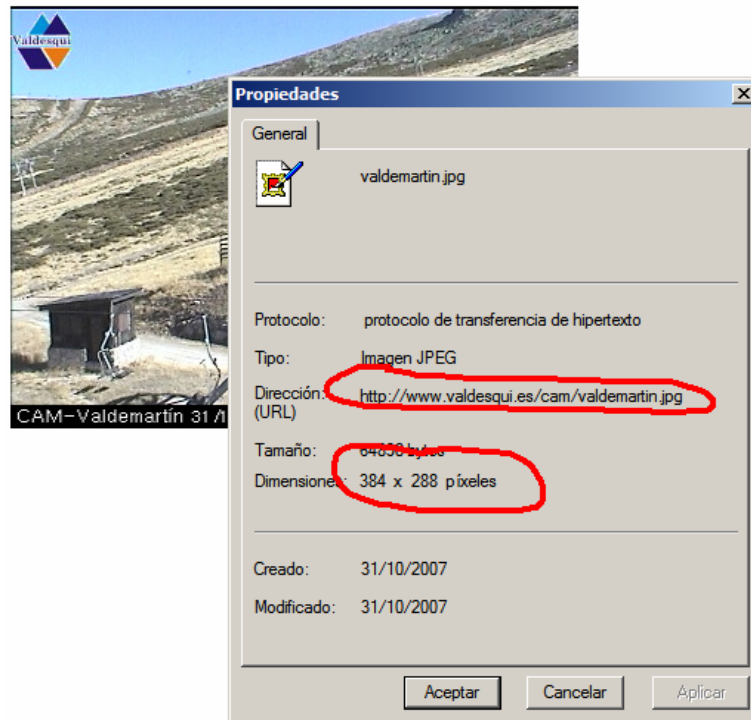
var allowHome = true

</script>
</body>
</html>
```

The original image, extracted from the URL <http://www.valdesqui.es/cam/valdemartin.jpg> would be as follows:



The real size is noticeable bigger than QCIF, as you can see:



The result on the screen of a mobile phone accessing an application PIMobile showing the example page is like the one shown here.

As can be seen, shows the content tailored to the mobile screen, as requested in the Javascript code of the page (**startFit = true**)



It would also be possible to display the image at its actual size, but focusing on certain coordinates. This can be useful for example in a map. To do so would be

StartFit = false

X = ...

Y = ...

Where x and y are the coordinates where visualization should focus

In the example shown x = 100, y = 200



The javascript file:

```
<script src=scroll_zoom.js></script>
```

It contains the code needed for managing keyboard, and offers the following possibilities (numerical as befits the DTMF digits)

3, 7 = Zoom +/-

2,8,4,6 = Up/Down/Left/Right (scroll of the Picture if not fitted in the screen)

0 = Help (shows commands)

9 = Quit (navigate to another URL)



The number of pixel for zoom or scroll in each increment or movement can be determined in the HTML code.

Another possibility is to handle an object, for example a Media Player. If the object supports resizing through Javascript in the usual manner (by altering their properties width and height) it will be similar to the behaviour studied for the example above of the still webcam. The address:

<http://www.pimobile.es/samples/demoscrollobject.htm>

You can see an example of the webcam of a highway. It's enough to make the following changes in the code of the previous example:

```

        // URL of the image to show
        var imageURL = 'http://www.valdesqui.es/cam/valdemartin.jpg'
        document.write('<br><br><br>' )
with
        // URL of the image to show
        var imageURL = 'mms://stream2.terra.es/wm2live2'
document.write(' <object classid="CLSID:6BF52A52-394A-11d3-B153-00C04F79FAA6" +
        'type="application/x-oleobject" +
        'id="foto" +
        'width="176" height="134" +
        'standby="Loading content..." +
        'codebase="http://activex.microsoft.com/activex/controls/mplayer/en/nsmp2inf.cab#Versio
n=5,1,52,701">' +
        '<param name="autoStart" value="true"/>' +
        '<param name="balance" value="0"/>' +
        '<param name="currentPosition" value="0"/>' +
        '<param name="enableContextMenu" value="false"/>' +
        '<param name="enabled" value="false"/>' +
        '<param name="fullScreen" value="false"/>' +
        '<param name="mute" value="false"/>' +
        '<param name="playCount" value="1"/>' +
        '<param name="rate" value="1.0"/>' +
        '<param name="uiMode" value="none"/>' +
        '<param name="windowlessVideo" VALUE="true">' +
        '<param name="URL" value="' + imageURL + '">' +
        '<param name="volume" value="70"/>' +
        '</object>' )

```

EMULATORS

It is possible to use emulators to display on a PC what would be seen in a 3G mobile accessing an application PIMobile. Currently there are two that can be helpful, one for the development phase of the application and another for the deployment and complete verification.

It should not be forgotten that it is also possible to access any application PIMobile from a PC equipped with Internet Explorer just write the URL. In the upper left corner of your browser will display what would be on the phone.

PIMOBILE EMULATOR

Developed and supplied by CESTEL, it is an Internet Explorer browser with appearance of a mobile 3G and the screen size adjusted to the standard QCIF. You can write the URL to access, and click on the number keys for sending commands to the application.

It has the advantage of being able to test applications PIMobile without having to deploy in the PIMobile`s platform. In other words, applies especially in the stages of development prior to deployment, without any access to the platform.

To get this program contact the CESTEL PIMobile Technical Support.



X-LITE EMULATOR



Supplied by CounterPath

([Http://www.counterpath.com](http://www.counterpath.com)) is a softPhone meaning a software phone suitable for voice services over IP.

Through it is possible to access the address SIP (IP) platform where resides PIMobile and launch a video call, identical to the one conducted through a mobile phone 3G but without access through the network of a UMTS mobile service provider.

It has the advantage over the other simulator that lets you test the entire application, already installed in PIMobile platform and configured for access from a real mobile phone. Thus is most useful to verify the correct functioning of deployments and applications PIMobile

DESIGN RECOMENDATIONS FOR PIMOBILE APPLICATIONS

In addition to the screen size and the management of user commands (using numerical options and not links) take into account other recommendations, most of all visual, in the development of an application PIMobile.

- 1) The image shown in a mobile phone accessing a PIMobile is in truth a video coded using h.263 codec that is the standard for video calls. Due to the small band width and that the codec sacrifice image quality in exchange of less computing power (because of CPU cost and battery consumption on mobile phones), the images arriving to the phone are not all the sharp that could be desired. This makes needful to design using appropriate fonts and sizes looking for contrasts of colours that facilitate the readability of the image converted to H.263. It is not always easy and it is essentially a process of trial and error.
- 2) Las aplicaciones PIMobile aunque en realidad son aplicaciones web y en dicho campo no es frecuente el uso de audio a lo largo de la navegación, son accedidas mediante un sistema de videoconferencia. Por ello es recomendable que las páginas web accedidas

contengan asociadas locuciones, melodías o en general señales sonoras que acompañen la navegación del usuario para.

Applications PIMobile in truth are web applications and in that field is not frequent to use during navigation, but are accessed through a videoconferencing system. Therefore it is recommended that the web pages accessed contain associated audio phrases, melodies or generally sound signals that accompany user navigation. Use this to:

- a. Make more pleasant and attractive pages, making it easier to appreciate the superior multimedia capabilities of 3G compared to other systems such as WAP, not only in terms of visual possibilities but the hearing too.
 - b. Complementing the indications or visual information to avoid exhausting the limited space on the screen to communicate messages that can simply be heard. For example, to display an image with potential for scrolling or zooming, it could start to reproduce a phrase saying "Remember that you can press 0 to display the aid" or even "Remember that you can press 0 to listen aid"
- 3) The presentation and especially the capture of the information on screen should take into account the small space that is available. So in addition to always seek the shorter and simpler procedure, it will be necessary for certain complex sequences for capturing information (forms) or options (menus) are distributed in various screens in sequence.

DEPLOYMENT OF APPLICATION PIMOBILE

Applications PIMobile been a real Web applications do NOT RESIDE IN THE SERVICES PLATFORM WHERE IS LOCATED THE SERVER PIMobile, but in the Web servers hosting applications accessed via the URL of your pages.

Therefore the deployment of an application PIMobile consist of

- 1) Verify that the web servers that will be accessed have properly deployed the application.
- 2) Verify that these servers are accessible to the URLs, IP addresses and ports in question from the Internet, or at least from the PIMobile server IP.
- 3) Ask CESTEL an entry point from its content platform, providing the home URL of the application. That is, CESTEL, if verified that the application supports PIMobile, will assign a phone number for access to the URL in question, and define the number of concurrent connections allowed.

Once put into operation, it should be noted that the availability of an application PIMobile and their response time, therefore, depends not only on services platform and PIMobile, but of the web server that actually contains the application PIMobile.