

OPERA

OPERAscript

User & Developer Manual

Contents

A. Introduction

B. Programming Guide by examples

i. The fundamentals

ii. notfound.html

iii. embed_player.html

iv. external_player.html

v. setup.html

C. Appendix A. (Function description)

D. Appendix B. (Current limitation)

A. Introduction to opera.js

Opera.js is a JavaScript function library, which provides a set of functions to help web designers to integrate OPERA System into their webpages.

The Opera.js provides functions like detecting players, invoking players, setting cookies and etc. It helps web programmers, who do not know much about OPERA, to make use of the OPERA functions easier. The following chapters will explain what the functions do and how you can include them in your HTML code.

B. Programming guide by examples

i. The fundamentals

```
<HTML>
<body bgcolor=white text=black link=blue>
<SCRIPT LANGUAGE="JavaScript" src="opera.js"></script>
<table border=0>
<tr>
<td>Sorry, the <script>OPERA_PrintCriteriaValue("uml")</script> cannot be not found.
</td>
</tr>
<tr>
<td><a href="http://www.hkci.org" target=_new></a></td>
</tr>
</table>
</BODY>
</HTML>
```

This section will tell you how to include our javascript library in your HTML code. This is the fundamental of using our functions for building your Web.

Whenever you want to use our functions to perform OPERA functionality, you should include our js file between the BODY tag as shown in the figure above. i.e.

```
<script language="JavaScript" src="opera.js" ></script>
```

It should be included before you call any opera.js functions, otherwise, some unexpected results will occur.

The opera.js not only includes the definitions of all opera functions, it will also gets some of the important information which will be used in the included functions.

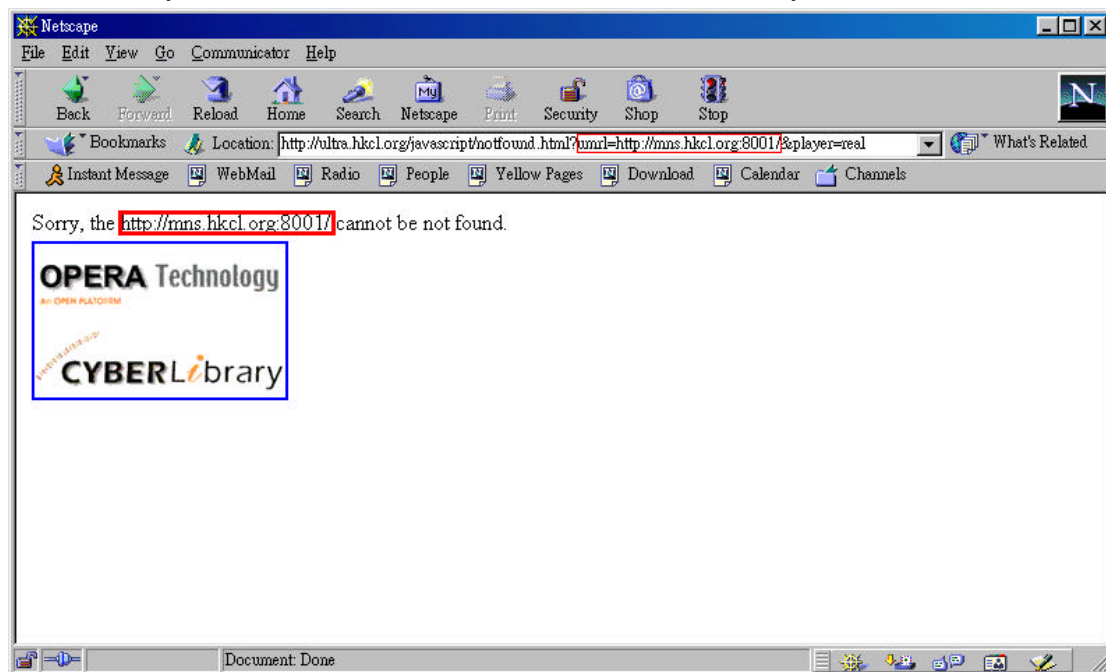
ii. notfound.html

The notfound.html is our first html example. We will give you some ideas of what the opera.js does after inclusion and introduce the OPERA_PrintCriteriaValue() function to you.

```
<HTML>
<body bgcolor=white text=black link=blue>
<SCRIPT LANGUAGE="JavaScript" src="opera.js"></script>
<table border=0>
<tr>
<td>Sorry, the <script>OPERA_PrintCriteriaValue("umrl")</script> cannot be not found.
</td>
</tr>
</table>
<td><a href="http://www.hkcl.org" target=_new></a></td>
</tr>
</table>
</BODY>
</HTML>
```

* The opera.js is included before the OPERA_PrintCriteriaValue() is called.

When you view the above html file in the browser, you will see,



In fact, after the inclusion of opera.js, it will try to get the current URL in the browser and create an associative array to store the value of the argument after the character ' ? ' , just like CGI argument.

For example,

URL:

<http://www.hkcl.org/notfound.html?umrl=http://mns.hkcl.org:8001/&player=real>

The opera.js will create an array to store the respective values of umrl and

player in the current URL of the browser.

The OPERA_PrintCriteriaValue() function is provided for you to print the value of the created associative array.

In the above example,

if you call OPERA_PrintCriteriaValue("umrl"), it will print "<http://mns.hkcl.org:8001/>" in the browser.

if you call OPERA_PrintCriteriaValue("player"), it will print "real" in the browser.

iii. embed_player.html

This html will invoke a player in the browser to play a video and create a small table to show the player used, the language and the bandwidth of the video.

```
<html>
<body bgcolor=black text=lightyellow vlink=yellow link=yellow alink=red>
<noscript>Please enable javascript</noscript>
<SCRIPT LANGUAGE="JavaScript" src="opera.js"></script>
<center>

<table border=0>
<tr>
<td>
<center>
<a href="http://www.hkcl.org" target=_new></a>
</center>
</td>
</tr>
<tr><td>
```

```
<script language=javascript>
var d_width=0;
var d_height=0;

switch(OPERA_CriteriaValue("Bandwidth")){
  case "45Mbps" : d_width = 320;
                  d_height = 240;
                  break;
  case "1.5Mbps" : d_width = 320;
                  d_height = 240;
                  break;
  case "512Kbps" : d_width = 320;
                  d_height = 240;
                  break;
  case "256Kbps" : d_width = 320;
                  d_height = 240;
                  break;
  case "128Kbps" : d_width = 320;
                  d_height = 240;
                  break;
  case "56Kbps" : d_width = 160;
                  d_height = 120;
                  break;
  case "28.8Kbps" : d_width = 160;
                   d_height = 120;
                   break;
  default: d_width = 160;
           d_height = 120;
           break;
};

if(OPERA_CriteriaValue("Player") == "QuickTime" &&
  OPERA_CriteriaValue("Bandwidth") == "256Kbps"){
  d_width = 400;
  d_height = 350;
};
```

```

</script>

<center>
<script language=javascript>
//EmbedPlayer|player, width, height, object str, param str, embed str, quicktime dummy mov);
if(OPERA.CriteriaValue("Player") == "MediaService"){
    OPERA_EmbedPlayer|"MediaService", d_width, d_height,
    'ID=NSPlay classid="CLSID:22D6F312-B0F6-11D0-94AB-0080C74C7E95"
    codebase="http://activex.microsoft.com/activex/controls/mpplayer/en/nsmp3inf.cab
    #Version=5,1,52,701" standby="Loading Microsoft Windows Media Player components..."',
    '<param name="ShowControls" value=0> <param name="ShowDisplay" value=0>
    <param name="ShowStatusBar" value=0> <param name="AutoSize" value=0>',
    'pluginspage="http://www.microsoft.com/Windows/Downloads/Contents/Products/MediaPlayer/"
    showcontrols=0 showdisplay=0 showstatusbar=0 type="video/x-ms-asf" name=NSPLAY ','');
};

if(OPERA.CriteriaValue("Player") == "Real"){
    OPERA_EmbedPlayer|"Real", d_width, d_height,
    'ID=RVOCX CLASSID="clsid:CFCDA03-8BE4-11cf-884E-0020AFB8C0FA"',
    '<PARAM NAME="CONTROLS" VALUE="ImageWindow"> <PARAM NAME="AUTOSTART" VALUE="true">',
    'type="audio/x-pn-realaudio-plugin" controls="ImageWindow" autostart="true",');
};

if(OPERA.CriteriaValue("Player") == "QuickTime"){
    OPERA_EmbedPlayer|"QuickTime", d_width, d_height, '', '',
    'type="video/quicktime" autoplay=true controller=true target="quicktimeplayer"
    bgcolor="black",opera.mov');
};
</script>
</center>

</td></tr>
<td>

</td></tr>
<td>
<center>
<table border=1>
<tr><td bgcolor=#000099>Format</td><td bgcolor=#000099>Language</td><td bgcolor=#000099>Bandwidth</td></tr>
<tr><td><b><script>OPERA_PrintCriteriaValue("Player");</script></b></td>
<td><b><script>OPERA_PrintCriteriaValue("Language");</script></b></td>
<td><b><script>OPERA_PrintCriteriaValue("Bandwidth");</script></b></td></tr>
<tr><td colspan=3>
<center>
<a href="javascript:OPERA_SetUpLink();">Re-select Your Personal Choice</a>
</center>
</td></tr>
</table></center></td>
</tr></table>
</center>
</body>
</html>

```

When you view the above the html in your browser, you will see,



As mentioned in the section, The fundamentals, the opera.js will store the value of the arguments of the current URL in an associative array. In this html, we assume that there are 3 arguments, player, language and bandwidth.

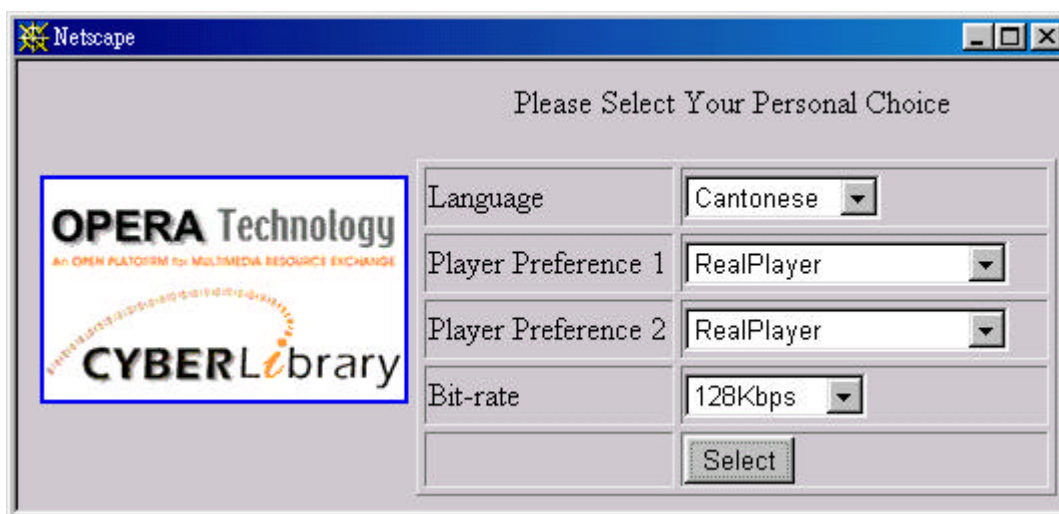
The URL of the browser in the above example will be like,

http://www.hkcl.org/embed_player.html?player=Real&language=Cantonese&bandwidth=56Kbps

It first checks what player will be used. Then the OPERA_EmbedPlayer() function will write <object> <embed> tags in html for the browser to invoke a player to play. Afterwards, it will call the OPERA_PrintCriteriaValue() function, which will print the value of an argument in associative array, to create the table shown above

Since difference bandwidth video and different players occupies different embedded space, it is necessary to pre-set the width and height of the embedded space. It is done by assigning values to d_width and d_height in the above example.

Lastly, the OPERA_SetUpLink() function will pass an argument to tell the MNS server that the user would like to change his/her preference. When you clicked the above link, you will see the setup table as shown below,



* This setup table will be explained in the v. setup.html.

iv. external_player.html

Actually, the external_player.html acts like nearly the same as embed_player.html, except it will invoke the player in a new window to play the video instead of embedding in the browser.

```

<html>
<body bgcolor=black text=lightyellow vlink=yellow link=yellow alink=red>
<noscript>Please enable javascript</noscript>
<SCRIPT LANGUAGE="JavaScript" SRC="opera.js"></script>

<table border=0><tr>
<td><a href="http://www.bkcl.org" target=_new</a></td>
<td><table border=1>
<tr><td bgcolor=#000099>Format</td><td bgcolor=#000099>Language</td><td bgcolor=#000099>Bandwidth</td></tr>
<tr><td><script>OPERA_PrintCriteriaValue("player");</script></b></td>
<td><b><script>OPERA_PrintCriteriaValue|"language";</script></b></td>
<td><b><script>OPERA_PrintCriteriaValue|"bandwidth";</script></b></td></tr>
<tr><td colspan=3>
<center>
<a href="javascript:OPERA_GetUplink();">Re-select Your Personal Choice</a>
</center>
</td></tr>
</table></td>
</tr></table>
</body>
</html>

```

v. setup.html

The setup.html is our last example of using opera.js and in fact, it covers the most important application of opera.js and is the most complicated.

The setup.html will be called when you clicked the link in the embed_player.html and external_player.html. The table below is generated by the setup.html.



The setup.html is as follows:

```
<HTML>
<body bgcolor=#CCCCCC>
<SCRIPT LANGUAGE="JavaScript" src="opera.js"></script>
<SCRIPT LANGUAGE="VBScript" src="opera.vbs"></script>
<SCRIPT LANGUAGE="JavaScript" src="local.js"></script>
<table border=0>
<tr>
<td><a href="http://www.hkci.org" target=_new></a></td>
<td>
<center>Please Select Your Personal Choice
<form method=GET name=vwform OnSubmit="return OPERA_Check(this);">
<script>OPERA_SetUpFormHiddenField();</script>
<table ID="Table2" BORDER="1" CELLSPACING="3" CELLPADDING="1" WIDTH="320">
<script language="javascript">
OPERA_GetExistingUserPreference();
OPERA_MakeUserOption('Language',180,27,150);
OPERA_MakePlayerOption('Player',180,27,150);
OPERA_MakeUserOption('Bandwidth',180,27,150);
</script>
<tr>
<td>&nbsp;</td>
<td WIDTH=180><P><INPUT TYPE=submit NAME="FormButton1" VALUE="Select" ID="FormButton1">
</td>
</tr>
</table>
</td>
</tr>
</table>
</form>
</center>
</td></tr>
</table>
</BODY>
</HTML>
```

Besides including opera.js, the setup.html also include opera.vbs, a VBScript file. It is used to detect player in Internet Explorer. The syntax is:
<script language="VBScript" src="opera.vbs"></script>

Opera.vbs must be included after opera.js, otherwise, the detection of players in Internet Explorer will be failed.

Local.js is also included in setup.html, a Javascript file which include the information need to make options as bellows:

```
OPERA_CriteriaValue['Language'] = new Array();
OPERA_CriteriaValue['Language'][0] = 'Cantonese';
OPERA_CriteriaValue['Language'][1] = 'Putonghua';
OPERA_CriteriaValue['Language'][2] = 'English';
OPERA_CriteriaValue['Language'][3] = 'Japanese';

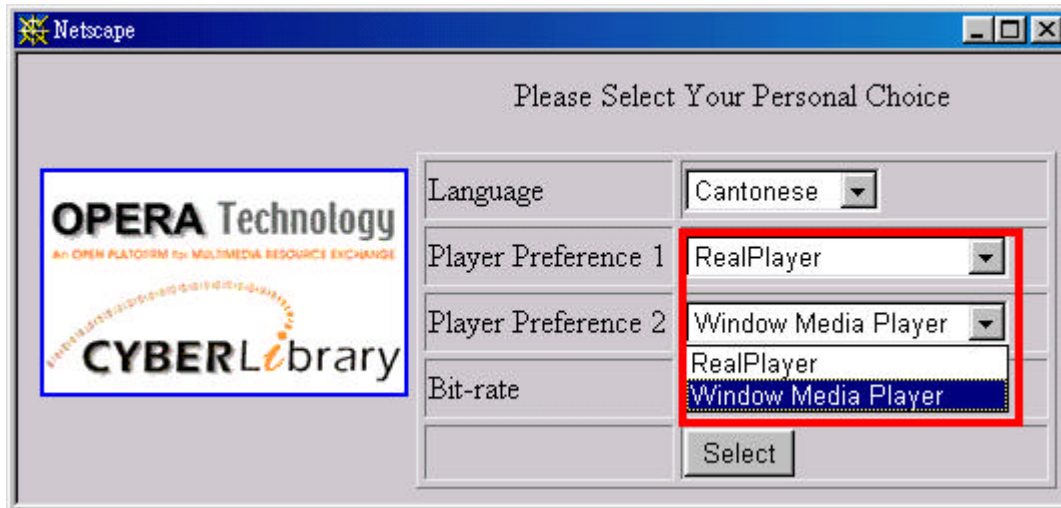
OPERA_CriteriaValue['Bandwidth'] = new Array();
OPERA_CriteriaValue['Bandwidth'][0] = '45Mbps';
OPERA_CriteriaValue['Bandwidth'][1] = '1.5Mbps';
OPERA_CriteriaValue['Bandwidth'][2] = '512Kbps';
OPERA_CriteriaValue['Bandwidth'][3] = '256Kbps';
OPERA_CriteriaValue['Bandwidth'][4] = '128Kbps';
OPERA_CriteriaValue['Bandwidth'][5] = '56Kbps';
OPERA_CriteriaValue['Bandwidth'][6] = '28.8Kbps';
```

Setup.html will then create a form for the user to select their preference. The OPERA_SetUpFormHiddenField() is called to create 2 hidden fields, i.e. org_umrl and re_config. This function should be called inside the form for normal operation.

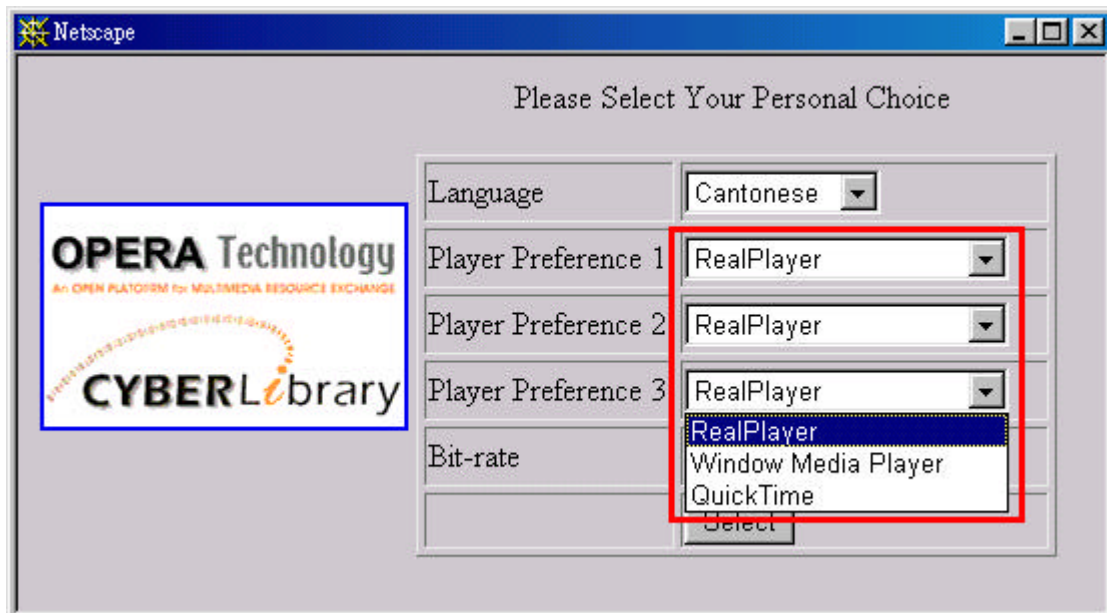
Afterwards, the OPERA_GetExistingUserPrefernece() is called in order to allow the OPERA Script to read the cookie in the browser to initialize the player preference values, which the user selected before, in the setup page. (This function is only necessary if you want the setup page to initialize the user previous player preferences.)

OPERA_MakePlayerOption () will detect what players(e.g RealPlayer, MS Media Player, QuickTime) are installed in the user's computer and create a group of select fields, depend on how many player installed.

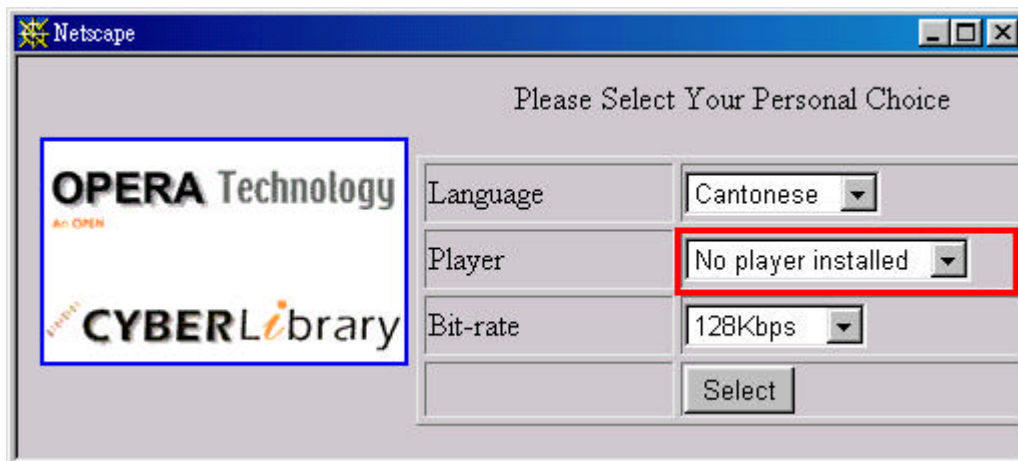
For example, if my computer only has RealPlayer and Window Media Player installed, there will only 2 select fields with 2 options in each:



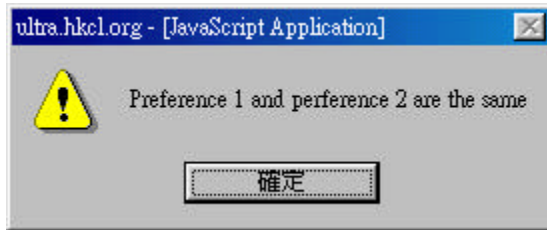
On the other hand, if my computer has also QuickTime installed, there will be 3 select fields with 3 options in each:



If the user has no player installed, the select field will be:



When the user press the select button in the above form, the OPERA_check() will be called to check whether the input is collect. If the user selects 2 players preferences with the same player, the following alert box will be popped up.



If everything is correct, the OPERA_check() will create a cookie to save the user' s preference.

C. Appendix A (functions description)

Global variable in the program

var OPERA_cgipara
(To store the browser current URL)

var OPERA_ptr = 0;
var OPERA_ptr1 = 0;
var OPERA_ptr2 = 0;
(temporary variables for breaking the arguments and store them into the array)

var opera = new Array();
(The created associative array)

var OPERA_i = 0;
(temporary variable)

var OPERA_RealPlayer = 0;
var OPERA_QuickTime = 0;
var OPERA_Media = 0;
(for determine which player is in the user computer, if the player is in user computer, it become 1)

var OPERA_gotReal = 0;
var OPERA_gotMedia = 0;
var OPERA_gotQuick = 0;
(for detection purpose in Internet Explorer,)

var OPERA_Cookie_array = new Array();
var OPERA_Player_array = new Array();
(for cookie operation of the opera.js)

OPERA_SetUpLink():

Syntax : OPERA_SetUpLink()

Description: The function prepare setup Link. It will help you to redirect your URL to setup.html so that users can re-select their own personal choice.

```
<html>
<body bgcolor=black text=lightyellow vlink=yellow link=yellow alink=red>
<noscript>Please enable javascript</noscript>
<script LANGUAGE="JavaScript" SRC="opera.js"></script>

<table border=0><tr>
<td><a href="http://www.hkcl.org" target=_new</a></td>
<td><table border=1>
<tr><td bgcolor=#000099>Format</td><td bgcolor=#000099>Language</td><td bgcolor=#000099>Bandwidth</td></tr>
<tr><td><script>OPERA_PrintCriteriaValue("player");</script></b></td>
<td><b><script>OPERA_PrintCriteriaValue|"language";</script></b></td>
<td><b><script>OPERA_PrintCriteriaValue|"bandwidth";</script></b></td></tr>
<tr><td colspan=3>
<center>
<a href="javascript:OPERA_SetUpLink();">Re-select Your Personal Choice</a>
</center>
</td></tr>
</table></td>
</tr></table>
</body>
</html>
```

External_player.html

OPERA_CriteriaValue() :

Syntax : OPERA_CriteriaValue(criteria)

PARAM: criteria

TYPE: String

The index of the created associative array

Description: It gets the value of the arguments in the array created by the opera.js. For example:

The URL is:

[http://www.hkcl.org/example.html?umrl=http://mns.hkcl.org:8001/&playe
r=real&language=Cantonese](http://www.hkcl.org/example.html?umrl=http://mns.hkcl.org:8001/&playe
r=real&language=Cantonese)

If you call OPERA_CriteriaValue("umrl"), it will return "<http://mns.hkcl.org:8001/>

If you call OPERA_CriteriaValue("player"), it will return "real".

If you call OPERA_CriteriaValue("Language"), it will return "Cantonese".

OPERA_PrintCriteriaValue():

Syntax: OPERA_PrintCriteriaValue(criteria)

PARAM: criteria

TYPE: String

The index of the created associative array

Description: It prints out the value of the input parameter. It is similar to OEPRC_CriteriaValue(), but it prints Criteria value, instead of returning Criteria value.

OPERA_SetCookieLink():

Syntax: OPERA_SetCookieLink(form)

PARAM: form

TYPE: form Object

The form that the users select their preferences

Description: The function runs suitable multimedia player, uses suitable language and bandwidth. (should add **SetUpFormHiddenField()**)

OPERA_EmbedPlayer():

Syntax: OPERA_EmbedPlayer(player, width, height, object_atr, param_atr, embed_atr, quicktime_dummy_mov)

PARAM: player

TYPE: String

The name of the player

PARAM: width

TYPE: Integer

The width in pixels of the embedded players

PARAM: height

TYPE: Integer

The height in pixels of the embedded players

PARAM: object_atr

TYPE: String

The class id of the player

PARAM: param_atr

TYPE: String

The parameters passed to the player

PARAM: embed_atr

TYPE: String

The attributes for the embedded player passed to the browser

PARAM: quicktime_dummy_movie

TYPE: String

The filename of the movie you want to play. Only used for QuickTime, for other players, pass a null string to this parameter

Description : It outputs the tag for creating the Embedded players.

OPERA_SetUpFormHiddenField():

Syntax: OPERA_SetUpFormHiddenField()

Description: It provides two hidden field name with (name="org_umrl" and name="re_config") and its value(opera["org_umrl"] and opera["re_config"]). It should be included in the setup form for normal operation.

OPERA_DetectPlayer():

Syntax: OPERA_DetectPlayer()

Description: The function detects Multimedia players that the client computer have. If the computer have the player, then the function will set the variable OPERA_player= 1, otherwise OPERA_player = 0. for example:

For a computer it have RealPlayer and Window Media Player, then

OPERA_RealPlayer = 1;

OPERA_Media = 1

OPERA_QuickTime = 0

OPERA_GetExistingUserPreference ():

Syntax: OPERA_GetExistingUserPreference ()

Description: Get the OPERA Cookie in the browser and put them into appropriate global array for further user-setup page initialization. (i.e. if you want the user setup page initialize the value of the user who has set before, you should call this function before any selection-creating functions in the following called)

OPERA CreateOpt():

Syntax: OPERA_CreateOpt(selectname, select_id)

PARAM: selectname

TYPE: String

The select name in the select field of the form. i.e.

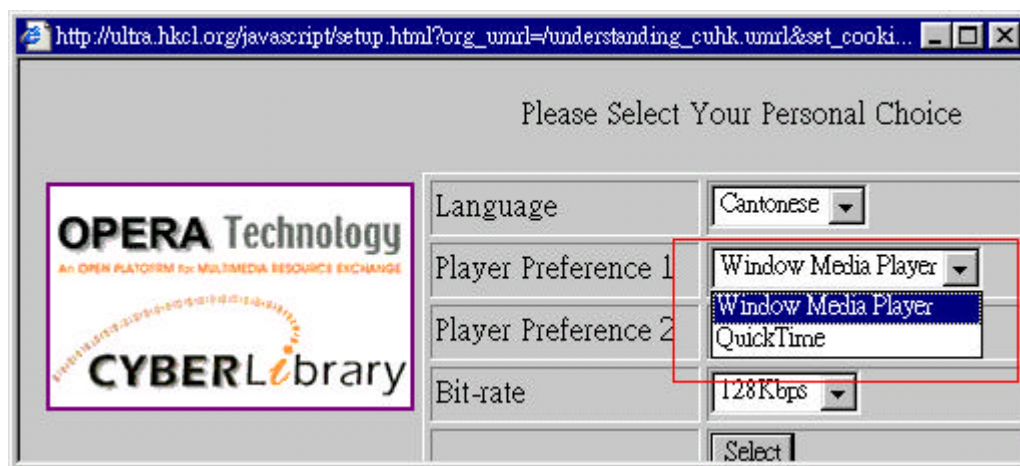
<select name = selectname> in html

PARAM: select_no

TYPE: Integer

The number id of this selection field

Description: The function provides a selection box for the user to choose suitable multimedia player. It only provides multimedia player option that user computer has(the function include DetectPlayer()). For example, if the users have only Window Media Player and Quicktime, then the selection box is as below:



OPERA MakePlayerOption ():

Syntax: OPERA_Pref(select_name , select_width, select_height, option_width)

PARAM: select_name

TYPE: String

The select name in the select field of the form. i.e.

<select name=select_name> in html

PARAM: select_width

TYPE: Integer

The width for the title. i.e. the width of “Player Preference #” in the below figure

PARAM: select_height

TYPE: Integer

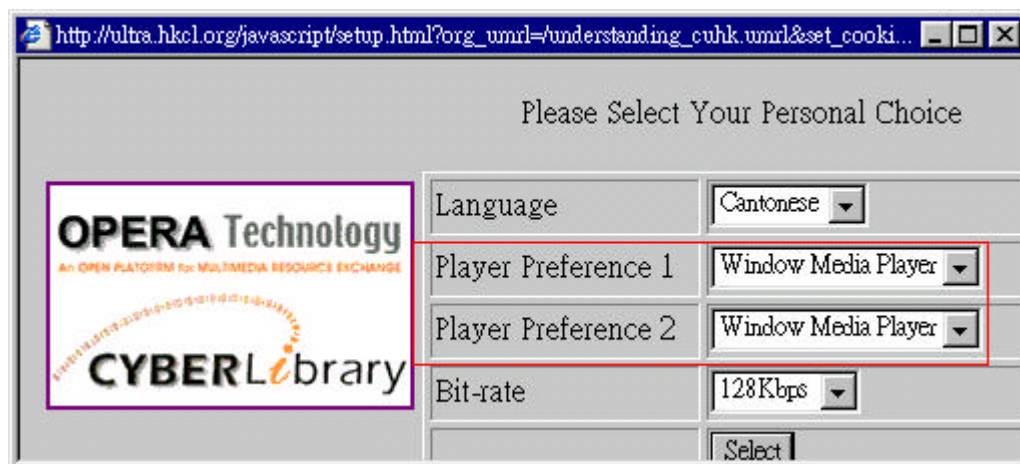
The height for the title. i.e. the height of “Player Preference #” in the below figure

PARAM: option_width

TYPE: int

The width for the player selection field.

Description: The function provides N selection box, where N is equal to the number of different multimedia players detected in the user computer. For example if the user have only Window Media Player and Quicktime, then N is equal to two, so it have two selection box (Player Preference 1 and Player Preference 2) for user to choose as below:



OPERA_CheckCookie():

Syntax: OPERA_CheckCookie(array, index, check_value)

PARAM: array

TYPE: Array Object

Specify the Global Array for search

PARAM: index

TYPE: String or Integer

The index of the array to search

PARAM: check_value

TYPE: String

The value being checked

Return: A String "SELECTED" if the check result is true or a null string otherwise

Description: This function is for internal use to initialize the user-setup page value

OPERA_Check():

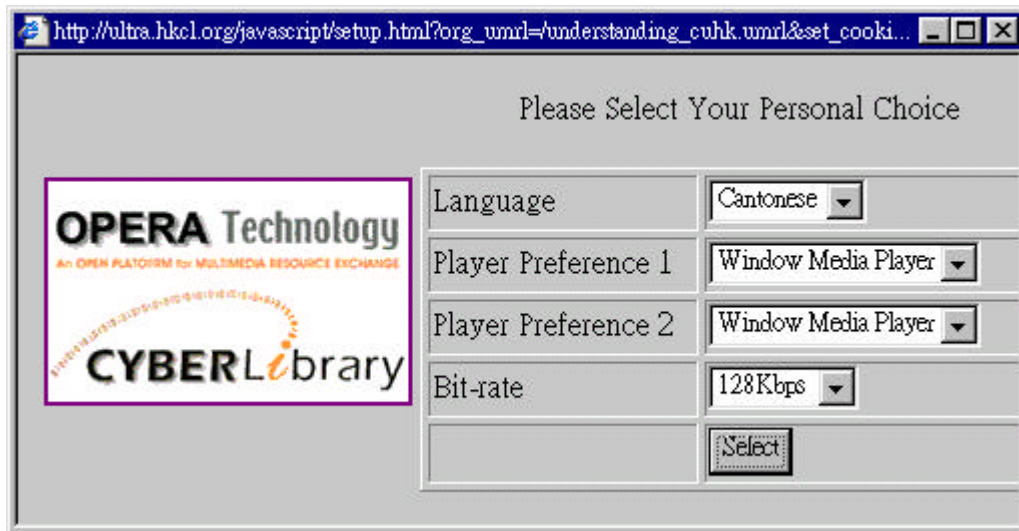
Syntax: OPERA_Check(form)

PARAM: form

TYPE: form Object

The form that the users select their preferences

Description: The function is used to check whether the player have chosen different Player in different selection box, if the user have chosen same Player in any two selection box then after pressing Select,



There will have message as below:



Then user have to choose again so that different Player is in different selection box.

If input is correct, the function will have action to submit the form for further process.

OPERA MakeUserOption():

Syntax: OPERA_Pref(select_name , select_width, select_height, option_width)

PARAM: select_name

TYPE: String

The select name in the select field of the form. i.e.

<select name=select_name> in html

PARAM: select_width

TYPE: Integer

The width for the title. i.e. the width of “select_name” in the below figure

PARAM: select_height

TYPE: Integer

The height for the title. i.e. the height of “select_name” in the below figure

PARAM: option_width

TYPE: int

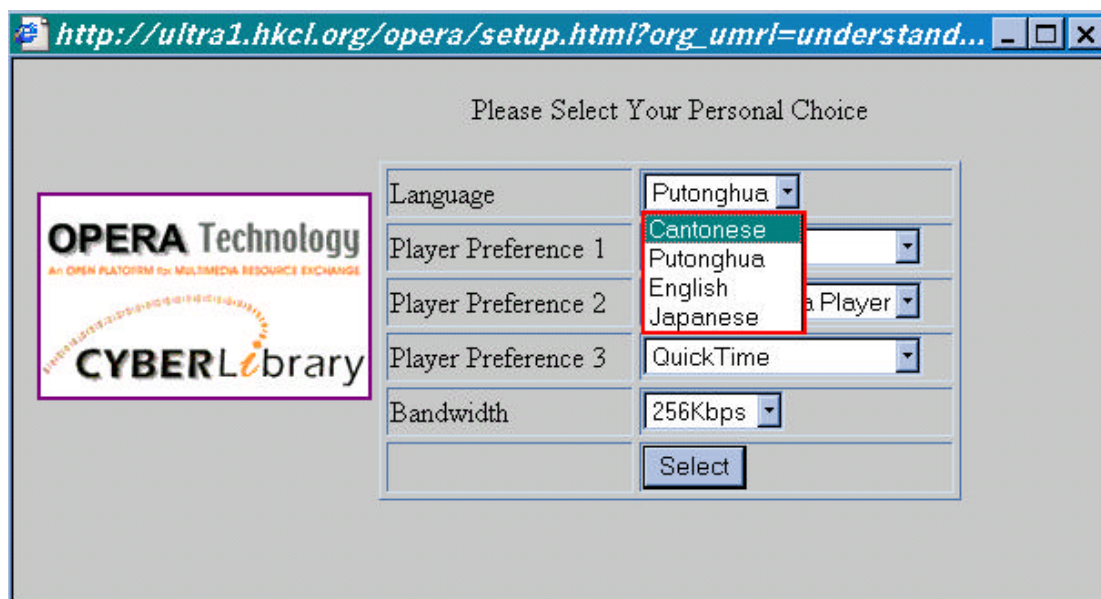
The width for the player selection field.

Description: The function provides 1 selection box for the select_name, where the options read from local.js. for example, in local.js:

```
OPERA_CriteriaValue[ 'Language' ] = new Array();  
OPERA_CriteriaValue[ 'Language' ][0] = 'Cantonese';  
OPERA_CriteriaValue[ 'Language' ][1] = 'Putonghua';  
OPERA_CriteriaValue[ 'Language' ][2] = 'English';  
OPERA_CriteriaValue[ 'Language' ][3] = 'Japanese';  
  
OPERA_CriteriaValue[ 'Bandwidth' ] = new Array();  
OPERA_CriteriaValue[ 'Bandwidth' ][0] = '45Mbps';  
OPERA_CriteriaValue[ 'Bandwidth' ][1] = '1.5Mbps';  
OPERA_CriteriaValue[ 'Bandwidth' ][2] = '512Kbps';  
OPERA_CriteriaValue[ 'Bandwidth' ][3] = '256Kbps';  
OPERA_CriteriaValue[ 'Bandwidth' ][4] = '128Kbps';  
OPERA_CriteriaValue[ 'Bandwidth' ][5] = '56Kbps';  
OPERA_CriteriaValue[ 'Bandwidth' ][6] = '28.8Kbps';
```

It define there is a selection name Language with options Cantonese, Putonghua, English and Japanese.

Then the selection box will become:



which has four options which defined in local.js.

D. Appendix B (Current limitations)

Detection of player in windows 98

Players/Bowser	Netscape	Internet Explorer
Real Player	OK	OK
Window Media Player	OK	OK
Quicktime	OK	OK

Detection of player in Macintosh

Player/Brower	Netscape	Internet Explorer
Real Player	OK	Not OK
Window Media Player	OK	Not OK
Quicktime	OK	Not OK

For Internet Explorer, as it cannot detect which multiplayer the user has, We provide all multimedia players for user to choose.

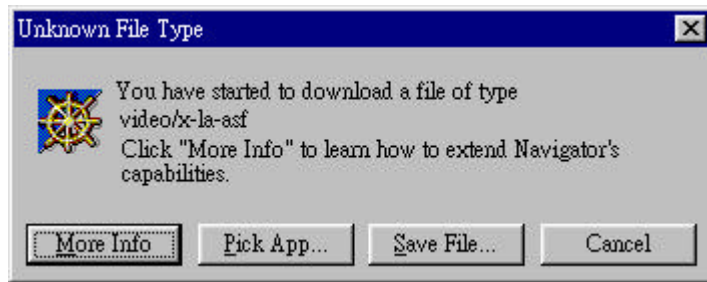
Detection of player in window 2000

Players/Bowser	Netscape	Internet Explorer
Real Player	OK	OK
Window Media Player	OK (But Problem In Accessing Player, solution below)	OK for window media player 7 Not OK for window media player 6.4 or before
Quicktime	OK	OK

For Window Media Player, as Window 2000 cannot detect if version is 6.4 or before (Microsoft suggest that users should install Version 7.0 for stability) We assume users use Window Media Player 7.0 if they have Window Media Player.

Accessing Window Media Player problem in Win2000

1. If you choose window media player for viewing video, there will have problem below:



You should choose Pick Application and install Mime(video/x-la-asf), Win2000 can detect Window Media Player in Netscape.