

HTA – HOW CAN I LET USERS CHOOSE A DEFAULT PRINTER FROM A LIST OF PRINTERS?

This is a way (from **Hey Scripting Guy!**) for users to choose a default printer from a list of the printers installed on their computer:

```
<SCRIPT Language="VBScript">
Sub Window_Onload
strComputer = "."
Set objWMIService = GetObject("winmgmts:\\\" & strComputer & "\root\cimv2")
Set colPrinters = objWMIService.ExecQuery("Select * From Win32_Printer")
For Each objPrinter in colPrinters
strPrinter = objPrinter.Name
Set objOption = Document.createElement("OPTION")
objOption.Text = strPrinter
objOption.Value = strPrinter
AvailablePrinters.Add(objOption)
Next
End Sub
Sub SetDefault
strPrinter = AvailablePrinters.Value
Set WshNetwork = CreateObject("Wscript.Network")
WshNetwork.SetDefaultPrinter strPrinter
Msgbox strPrinter & " has been set as your default printer."
End Sub
</SCRIPT>
<select size="5" name="AvailablePrinters"></select><p>
<input type="button" value="Set as Default" onClick="SetDefault">
```

Let's talk about what we have here; what we *do* have here is an **HTA (HTML Application)**. HTAs are simply a way for script writers to create graphical user interfaces; they are essentially Web pages (albeit with a .hta file extension) designed to run off your local machine. (*Yes, we know: that's not a very good explanation. If you need more information about HTAs, well, that's what the [HTA Developers Center](#) is for*)

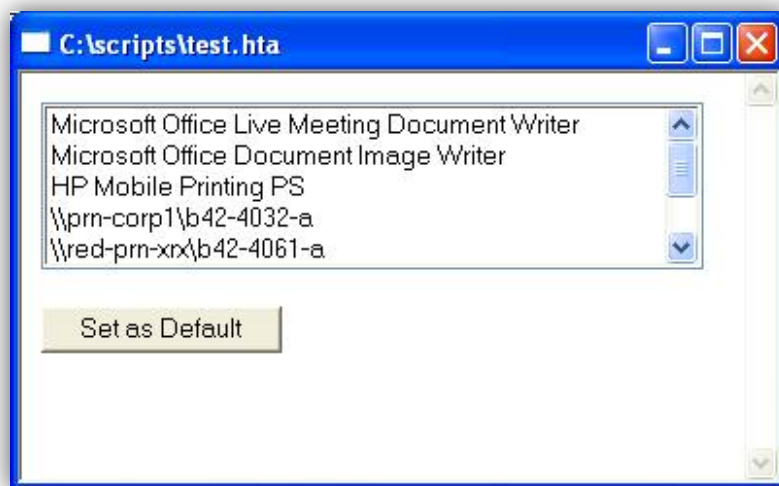
VBScript doesn't have a native method for creating graphical user interfaces.

That's a problem, but we can work around that problem by creating an HTA, and then calling that HTA from within our logon script.

And that's exactly what we've done here: we've created a .hta file, a file that we'll need to copy to each of our computers. (But that's no problem; your logon script can do that.) Inside our logon script we then need just two lines of code to pop up this HTA and, in turn, let users select a default printer:

```
Set objShell = CreateObject("Wscript.Shell")
objShell.Run "C:\Scripts\Test.hta"
```

So what will people see when this HTA pops up? They'll see – in all its barebones splendor – something that looks like this:



Yes, we know. But if you want beautiful you're on your own.

Now that you understand the basic scenario let's take a closer look at the HTA itself. As far as the HTML tagging goes, well, there really isn't much: all we have is a list box named *AvailablePrinters* (added to the HTA by using the **<SELECT>** tag) and a button (**<INPUT TYPE = "BUTTON">**):

```
<select size="5" name="AvailablePrinters"></select><p>
<input type="button" value="Set as Default" onClick="SetDefault">
```

The idea here is that you select a printer from the list box and then click the button; upon doing so, the printer you selected will be configured as your default printer.

That's right: just like magic.

If you've done much HTML tagging you probably noticed that, as configured, our list box is empty; it doesn't include a single installed printer. Why not? Well remember, that's the whole idea here: we need a dynamic list box, one that, each time the HTA starts, gets populated with the printers (and only those printers) installed on the computer in question. How do we propose to do *that*? Why, by using this subroutine, of course:

```
Sub Window_Onload
strComputer = "."
Set objWMIService = GetObject("winmgmts:\\\" & strComputer & "\\root\cimv2")
```

```
Set colPrinters = objWMIService.ExecQuery("Select * From Win32_Printer")
For Each objPrinter in colPrinters
strPrinter = objPrinter.Name
Set objOption = Document.createElement("OPTION")
objOption.Text = strPrinter
objOption.Value = strPrinter
AvailablePrinters.Add(objOption)
Next
End Sub
```

This, needless to say, is a subroutine named **Window_Onload**; in an HTA (or in a Web page) any subroutine with that name automatically runs whenever the HTA is opened or refreshed. Inside this subroutine, we first use this block of code to retrieve a collection of all the printers installed on the local computer:

```
strComputer = "."
Set objWMIService = GetObject("winmgmts:\\." & strComputer & "\root\cimv2")
Set colPrinters = objWMIService.ExecQuery("Select * From Win32_Printer")
```

That was easy, wasn't it? The question now is this: what do we *do* with this collection of printers?

To begin with, we set up a For Each loop to loop through each printer in the collection. For each printer we grab the value of the **Name** property and store it in a variable named **strPrinter**. We then run headlong into *this* line of code:

```
Set objOption = Document.createElement("OPTION")
```

What we're doing here is creating an instance of the HTML **Option** object; as the name implies, this object represents an individual option in a list box. After creating a "blank" option we then assign the printer name to the option's **Text** and **Value** properties:

```
objOption.Text = strPrinter
objOption.Value = strPrinter
```

In case you're wondering, the **Text** property represents the actual text displayed in the list box; the **Value** property is, well, the value assigned to that option. For this particular list box we made the **Text** and **Value** identical, but they don't have to be. For example, suppose you had a list box that allowed people to select US states.

You could set the **Text** of an option to *Washington*; that way people would see the name *Washington* in the list box. At the same time, however, you could set the value of that option to *WA*. That way people would see a "**friendly**" name in the list box, but you could still use the official postal abbreviation in your script.

After we assign the property values we then call the **Add** method and add the option (that is, the name of the first printer) to the list box:

```
AvailablePrinters.Add(objOption)
```

And then we repeat the process with the next printer in the collection.

The **Window_Onload** subroutine populates our list box with the names of all the printers installed on the local computer. To make one of these printers the default printer we need to select a the printer from the list box and then click the **Set as Default** button. Clicking the button causes the following subroutine to run:

```
Sub SetDefault
strPrinter = AvailablePrinters.Value
Set WshNetwork = CreateObject("Wscript.Network")
WshNetwork.SetDefaultPrinter strPrinter
Msgbox strPrinter & " has been set as your default printer."
End Sub
```

As you can see, we're not doing anything too-terribly fancy in this subroutine. To begin with, we grab the **Value** from the item selected in the list box and assign that to a variable named **strPrinter**:

```
strPrinter = AvailablePrinters.Value
```

We use these two lines of code to create an instance of the **Wscript.Network** object and then call the **SetDefaultPrinter** method in order to make the selected printer the default printer:

```
Set WshNetwork = CreateObject("Wscript.Network")
WshNetwork.SetDefaultPrinter strPrinter
```

And then, just for the heck of it, we display a message stating that the selected printer is now the default printer.

And that's basically all we have to do. Admittedly we don't have any error-handling here and we don't have an obvious way for users to close the dialog box without choosing a new default printer. Likewise we could get a little fancier and, say, display the name of the current default printer in a different color. But look at it this way: if we did *all* the work you might start to feel like no one needed you, and you might be tempted to jump off a bridge or throw yourself on a scrap heap.

Which means that all these things we don't do are actually something we do for your own good.

You're welcome.