How Can I List All the User Profiles on a Computer?

It's not that we have anything against user profiles. The problem is that user profiles represent an important part of a system administrator's duties, and because of that everyone assumes that there must be all sorts of really cool scripting technologies that make it easy to manage user profiles.

As it turns out, you can find a list of all the user profiles in the **HKEY_LOCAL_MACHINE** portion of the registry; more specifically, you can find them listed under this registry key:

```
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList
```

The bad news is that you don't get a nice, neat list of user profiles; instead, you see a series of separate registry subkeys, each one named after a user's **SID** (**Security Identifier**). To determine the actual user profiles found on a computer we need to iterate through each of those individual subkeys and read the value of **ProfileImagePath**; as shown in the following screenshot, **ProfileImagePath** gives us the path to the profile and, by extension, the name of the user that the profile belongs to:

So how do we iterate through all these subkeys and read the value of **ProfileImagePath**? Well, as we almost always do any time we're faced with a scripting issue that involves the registry, we turn to **WMI**:

```
On Error Resume Next
Const HKEY_LOCAL_MACHINE = &H80000002
strComputer = "."
Set objRegistry=GetObject("winmgmts:\\" & _
    strComputer & "\root\default:StdRegProv")
strKeyPath = "SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList"
objRegistry.EnumKey HKEY_LOCAL_MACHINE, strKeyPath, arrSubkeys
For Each objSubkey In arrSubkeys
    strValueName = "ProfileImagePath"
    strSubPath = strKeyPath & "\" & objSubkey
    objRegistry.GetExpandedStringValue
HKEY_LOCAL_MACHINE,strSubPath,strValueName,strValue
    Wscript.Echo strValue
Next
```

Don't worry: we'll explain how this all works. We begin by defining a constant named HKEY_LOCAL_MACHINE and assigning it the value &H80000002; we'll use this constant later on to indicate the part of the registry we want to work with. We then connect to the WMI service, specifying the path to the **StdRegProv** class (note that this is found in the **root\default** namespace).

After that we store the registry path (**SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList**) in the variable strKeyPath, then use the **EnumKey** method to return a collection of all the subkeys found

under **ProfileList** (as we've already determined, these subkeys - each one named after a user SID - represent all the user profiles found on the computer).

Note that the EnumKey method takes three parameters:

- HKEY_LOCAL_MACHINE, the constant that tells the script which portion of the registry to work with.
- strKeyPath, the variable containing the path to the ProfieList registry value.
- arrSubkeys, an "**out**" parameter that will hold the results returned by **EnumKey**. When we call EnumKey, the collection of registry subkeys found under **ProfileList** will be stored in arrSubkeys.

If we wanted to we could list all the subkey names by using code similar to this:

```
For Each objSubkey In arrSubkeys
     Wscript.Echo objSubkey
Next
```

So why don't we do that? Well, remember, these subkeys are named after user **SIDs**; simply listing the subkey names will return data similar to this:

```
S-1-5-18
S-1-5-19
S-1-5-20
S-1-5-21-1229272821-920026266-1060284298-1003
S-1-5-21-1229272821-920026266-1060284298-1005
S-1-5-21-1229272821-920026266-1060284298-1009
S-1-5-21-1454471165-1004336348-1606980848-8183
```

Interesting, but not very informative. (Unless, of course, you usually call people by their **SID** rather than by their name.)

Instead, what we need to do is connect to each of these registry subkeys and read the value of **ProfileImagePath**; that's what we do with these lines of code:

```
For Each objSubkey In arrSubkeys
    strValueName = "ProfileImagePath"
    strSubPath = strKeyPath & "\" & objSubkey
    objRegistry.GetExpandedStringValue
HKEY_LOCAL_MACHINE,strSubPath,strValueName,strValue
    Wscript.Echo strValue
Next
```

No, it's *not* too hard. We start off by defining a variable named strValueName and assigning it the value **ProfileImagePath**; this is the name of the individual registry value we want to read.

We then take our original registry path (**SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList**) and tack on a \ and the name of the first subkey in our collection (for example, S-1-5-18). That will give us a path to that first subkey, something similar to this:

```
SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList\S-1-5-18
```

Now that we know the path to the registry value we can use the **GetExpandedStringValue** method to read the value of **ProfileImagPath**. Why **GetExpandedStringValue**? Well, **ProfileImagePath** happens to be an expanded string value; it contains information similar to this:

```
%SystemDrive%\Documents and Settings\TestUser
```

The **GetExpandedStringValue** method will automatically replace a variable like *%SystemDrive%* with the actual value (e.g., C:). Thus we end up with output similar to this:

```
C:\Documents and Settings\TestUser
```

As long as we're on the subject, note that GetExpandedStringValue requires four parameters:

- **HKEY_LOCAL_MACHINE**, the constant that - again - represents the **HKLM** portion of the registry.
- **strSubPath**, the path to the first user profile subkey (for example, SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList\S-1-5-18).
- **strValueName**, a variable representing the registry value (**ProfileImagePath**) we want to read.
- **strValue**, another out parameter, this one containing the value of ProfileImagePath. The last thing we do in our For Each loop is echo the value of this out parameter, which will coincide quite nicely with the path to the user profile.

When we run the script we'll get back data similar to this:

```
C:\WINDOWS\system32\config\systemprofile
C:\Documents and Settings\LocalService
C:\Documents and Settings\NetworkService
C:\Documents and Settings\kenmyer
C:\Documents and Settings\TestUser
C:\Documents and Settings\packerman
```

Much better.

Incidentally, if all we wanted was the user name (for example, **kenmyer**) we could get fancy and use the VBScript **Split** function to separate the user name (the last item in the path) from the rest of the output.