Creating a new HDD using the Virtual Box COM Object

In an effort to learn a bit more about Virtual Box and creating tools for it I decided to start playing around with the Virtual Box SDK. After 45 min of trying to get it working in PowerShell natively I decided to make use of the Virtual Box COM Object instead – I will give the SDK another bash soon.

The first think I would like to try and automate is the ability to create a VHD (Virtual Hard Disk) which I would then be used to create a VM. So sounds simple enough, and with the documentation that the Virtual Box developers have created this was simple enough. You can see the documentation here.

After a bit of reading I now know I have the following steps to complete in order to create my new VHD:

1. Create an instance of the "VirtualBox.VirtualBox" COM Object
2. Call the "CreateHardDisk" method with the format and location of the disk
3. Call the "CreateBaseStorage" method with the size and storage method for the disk.

The only problem I had with using the SDK (or in my case the COM Object) is that I am unable to access any of the enums that I need to ensure that the disk type I am creating is the same no matter what version of the SDK I am using. If anyone knows how to do this I would be most appreciative to get the help. In the end I decided to resort to giving the "CreateBaseStorage" method a value of 1.

So my code looks something like this:

```
Clear-Host

$vbox = New-Object -ComObject VirtualBox.VirtualBox;
$drive_location = "c:\test-drive.vdi";
$drive_format = "vdi";
$drive_size = [System.UInt32] 256; # Size in Mb
$driveObj = $vbox.CreateHardDisk($drive_format, $drive_location);
$result = $driveObj.CreateBaseStorage($drive_size, 0);

if( $result.ErrorInfo -ne $null )
{
      "(!) There was an error creating your disk: "
      ""
      $result.ErrorInfo.Text
} else {
      "$drive_location was successfully created!"
}
```

And that's all there is to creating a VHD using the Virtual Box SDK and PowerShell. Keep reading on if you would like a bit of a breakdown of the script.

Script Breakdown

```
Clear-Host
```

Clear any output that might be in the console.

```
$vbox = New-Object -ComObject VirtualBox.VirtualBox;
```

Create a new instance of the “VirtualBox.VirtualBox” COM Object.

```
$drive_location = "c:\test-drive.vdi";
$drive_format = "vdi";
$drive_size = [System.UInt32] 256; # Size in Mb
```

Define the attributes of the disk I want to create. You may have noticed that the size of the disk is in MB, this is by design according to the documentation and makes sense. You might want to create a 1.5 GB disk (not likely) and working in Mb’s makes that a lot easier. The format for the disk is VDI (I am sure that any other format that Virtual Box supports can be used here).

```
$driveObj = $vbox.CreateHardDisk($drive_format, $drive_location);
```

By calling the “CreateHardDisk” method I am getting an object of type “IMedium” back, which I will call the “CreateBaseStorage” method on it to give the disk a size and storage method.

```
$result = $driveObj.CreateBaseStorage($drive_size, 0);
```

Here I am calling the “CreateBaseStorage” method to create the disk; I am saving the results of the creation to a variable which I will then check to ensure that the disk was created properly. The following lines are basic error checking to ensure that the disk was created properly.

```
if( $result.ErrorInfo -ne $null )
{
      "(!) There was an error creating your disk: "
      ""
      $result.ErrorInfo.Text
} else {
      "$drive_location was successfully created!"
}
```

NOTE: depending on the type of disk format you choose to create the “CreateBaseStorage” method could take a while to complete (if you choose to allocate all the space at once for example). There is an optional parameter on the “CreateBaseStorage” method that allows you to track the progress of the disk creation. See here for more information – this will populate an object of type IProgress with the current progress of the disk creation.