

October 2007

Converting C# code to Powershell syntax by sample

October 23, 2007

For the last few months I've been learning how-to create Powershell scripts to help manage our servers and build solutions for our users. First, I will show a sample C# code that will display a more complex messagebox in action. You can build a windows form with a button control and then add the C# code to test that this code works.

Here's the C# code:

```

//-----
// Sample C# code - Displaying MessageBox w/Yes-No buttons
//-----
string message = "Click on one of the Buttons with YES or NO";
string caption = "Sample w/YES-NO Button";
MessageBoxButtons buttons = MessageBoxButtons.YesNo;
DialogResult result;
DialogResult result2;

// Displays the MessageBox.
result = MessageBox.Show(message, caption, buttons);

if (result == System.Windows.Forms.DialogResult.Yes)
{
    result2 = MessageBox.Show("Your Answer is YES");
    // Closes the parent form.
    this.Close();
} else {
    result2 = MessageBox.Show("Your Answer is NO");
    // Closes the parent form.
    this.Close();
}

```

Here's the converted C# code to Powershell syntax:

```

###-----
### Converted C# code to Powershell syntax - Displaying MessageBox w/Yes-No buttons
###-----
[void] [System.Reflection.Assembly]::LoadwithPartialName("system.windows.forms") | out-null
$message = "Click on one of the Buttons with YES or NO";
$caption = "Sample w/YES-NO Button";

$result = [system.windows.forms.MessageBox]::Show($message,
    $caption, [System.Windows.Forms.MessageBoxButtons]::YesNo);

if ($result -eq [System.Windows.Forms.DialogResult]::Yes) {
    $result2 = [system.windows.forms.MessageBox]::Show("Your Answer is Yes");
    break;
} else {
    $result2 = [system.windows.forms.MessageBox]::Show("Your Answer is No");
    break;
}
}

```

Make sure to add this Powershell code to a *.ps1 file to run it. Or, just copy/paste code to the shell to run interactive. Now, you might asked the question, Can I build and convert a simple Windows form(s) solution to a Powershell syntax?

The answer is YES. Happy Powershell coding!