**Powershell Coding Standard**

**Last Modified: May 13, 2016 by Kris Sherrerd**

**5-13-2016 Initial creation**

***Standardization is Important***

**Names**:

**Class Names**

* Use a Specific name that makes sense for a Class Name
* Use Pascal Case for Class Names
* Shall not use all caps words
* Shall have no underscores or dashes ('\_', '-')

Name the class after what it is. If you can't think of what it is that is a clue you have not thought through the design well enough.

Suffixes are usually helpful. For example, if your system uses agents then naming something DownloadAgent conveys real information.

Example:

class JjLinkList

{

}

**Cmdlet & Function Name**

*A quick note. A True cmdlet is written and compiled, usually is C#, however we will simulate Cmdlets through the use of functions and for the purposes of this document they are one and the same.*

* Use a Specific Verb-Noun that makes sense for a function Name
* Use useful prefixes for function

Verb-Nouns used in function naming need to be very specific so that the user can discover your functions. Prefix generic nouns such as "server" with a shortened version of the product name. For example, if a noun refers to a server that is running an instance of Microsoft SQL Server, use a noun such as "SQLServer". The combination of specific nouns and the short list of approved verbs enable the user to quickly discover and anticipate functionality while avoiding duplication among function names.

* function name should be singular
* Use Pascal Case for function Names
* Use '-' as a Verb-Noun Separator

For example, use the name **Get-Process** instead of **Get-Processes**. It is best to follow this rule for all function names, even when a function is likely to act upon more than one item. Use Pascal case for parameter names. In other words, capitalize the first letter of verb and all terms used in the noun. For example, “Clear-ItemProperty”.

Variable Names

* Must use all lower case letters
* Must use '\_' as the word separator if does not contain a function
* Must use 'f\_' at the beginning of the name if it contains a function
* Do not start with a variable with '\_'

These should be descriptive, a noun if at all possible to describe what the variable is. We prefer all lower case to differentiate between functions/cmdlets and variables.

Example:

$alarm = ’12:00’

$f\_sed\_clock = cat DATA.TXT | % { $\_ -replace “6:00”, $alarm }

**Class Variable Names**

* my should be in front of any variables with no \_ following it and the beginning of the variable name
* Must use all lower case letters
* Must use '\_' as the word separator
* Do not start with a variable with '\_'

These names should have the word my in front of them, in order to distinguish that they are a part of the actual class when it is created. They also need to describe what information they will be holding if any. These are NOT to be used as a temp variable in any function.

**Cmdlet & Function Argument Names**

* Use Standard Parameter Names when possible

Please see <https://msdn.microsoft.com/en-us/library/dd878352%28v=vs.85%29.aspx> for the Microsoft standard names or see <https://github.com/stimepy/smallprgs/blob/master/Documents/Standard%20Cmdlet%20Parameter%20Names%20and%20Types.pdf> (Temporary Holding area!)

* Should use upper case letters and \_ as word separators, lower case for the rest of a word similar to Pascal Case for function Names

Should no standard parameter name be available, use the above to differentiate between a function argument variable and other variables.