POWERSHELL SCRIPT ANALYZER RULES

Fred Bainbridge

Fredbainbridge.com
@FredBainbridge

Wells Fargo

Jeff Scripter

Wells Fargo

▸ **Fred Bainbridge**	▸ **Jeff Scripter**
▸ @FredBainbridge	▸ Twitter Handle
▸ Microsoft MVP	▸ Awards, accomplishments, etc.
▸ 15 years	▸ Experience
▸ Beer and Baseball	▸ Favorite something; e.g., food

MMS

WHAT IS THE SCRIPT ANALYZER?

Who is using this today?

WHY THE SCRIPT ANALYZER?

Maintaining code quality is hard work.

MMS

Demo!

What comes out of the box?

https://github.com/fredbainbridge/CustomPSScriptAnalyzerRules

MMS

ANOTHER REASON WHY.

You can create your own rules!

MMS

Demo!

Function Naming Standards

https://github.com/fredbainbridge/CustomPSScriptAnalyzerRules

MMS

THE COMMUNITY RULES.

Yes it does.

https://github.com/PowerShell/PSScriptAnalyzer/tree/development/Tests/Engine/CommunityAnalyzerRules

MMS

Demo!

The Community Rules.

MMS

MORE REASONS WHY!

You can automate all of this. (Duh, its PowerShell)

Demo! Short Variable Names

$i see what you did there.

https://github.com/fredbainbridge/CustomPSScriptAnalyzerRules

MMS

EVEN MORE REASONS WHY!

You need to follow the rules to publish your code to the PowerShell Gallery.

MMS

CREATING YOUR OWN RULES

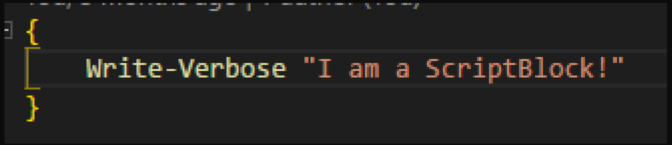
The fun stuff.

WHAT IS HAPPENING HERE?

- You are writing a module. (.psm1 file)
- Q: What is a script analyzer rule doing?
- A: It is taking a script block and testing it based on your conditions.

WHAT IS HAPPENING HERE? SCRIPTBLOCK EDITION

- What is a script block?



```
{
    Write-Verbose "I am a ScriptBlock!"
}
```

- Is it a string? No. It is a specific object type.

```
$ScriptBlock = [Scriptblock]::Create($string)
```

- https://docs.microsoft.com/en-us/dotnet/api/system.management.automation.scriptblock?view=powershellsdk-1.1.0

- The entirety of a ps1 file gets treated as a script block (generally) when using the script analyzer.

WHAT IS HAPPENING HERE? AST EDITION

- System.Management.Automation is a common namespace you will be working with.
- A script block is automatically converted to a collection of AST (Abstract Syntax Tree) objects. AST objects are the representation of your functions, variables, tokens, etc.
- You might be amazed to see how far and to what level of detail a script block gets broken down into.

https://docs.microsoft.com/en-us/dotnet/api/system.management.automation.language?view=powershellsdk-1.1.0

MMS

Breaking down a script into AST objects.

https://github.com/fredbainbridge/CustomPSScriptAnalyzerRules

MMS

WHAT IS HAPPENING HERE? AST FILTERING

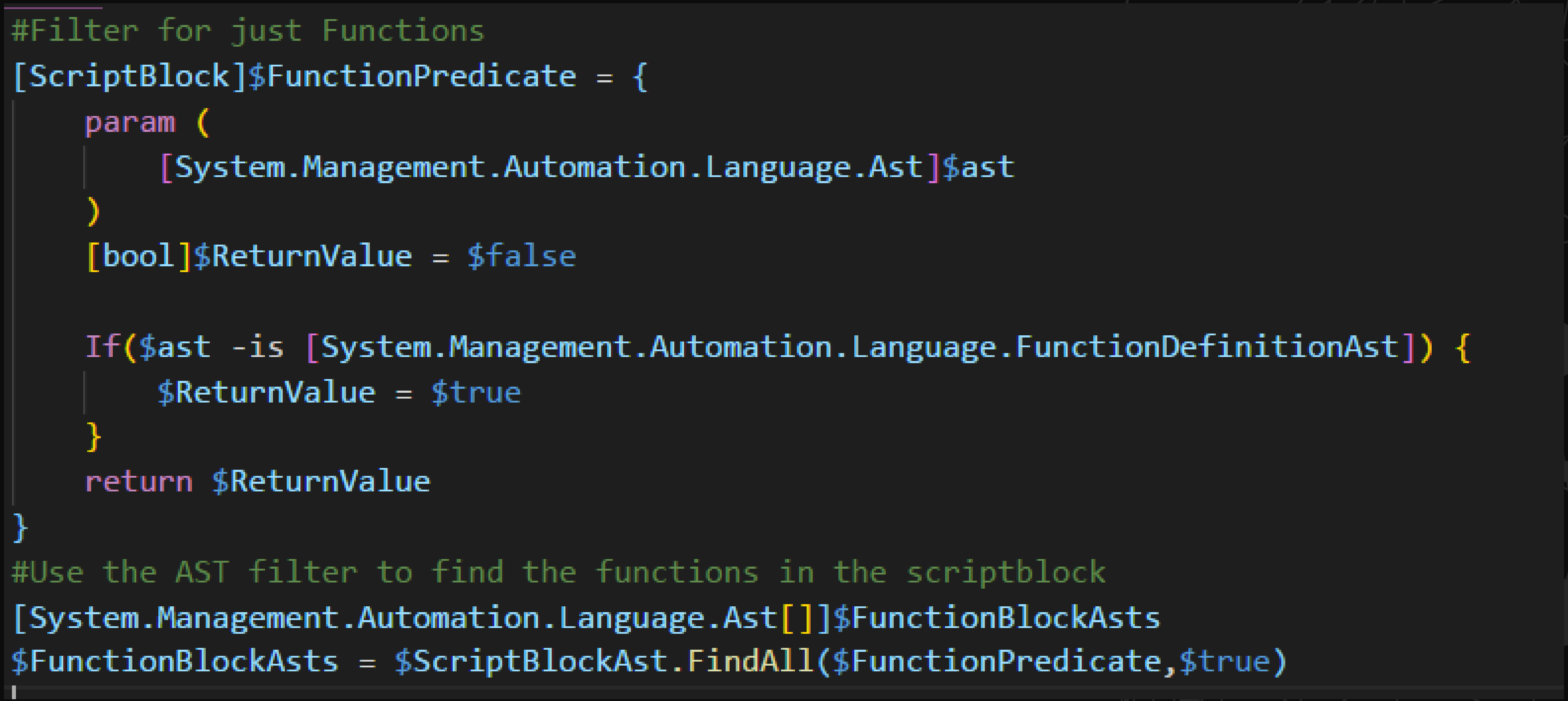
- You can then filter the Collection of AST objects to only return the ones you are interested in. i.e. Functions or a specific token.

- How can I find interrogate just the AST objects I care about?

Filtering AST objects. (Predicates)

https://github.com/fredbainbridge/CustomPSScriptAnalyzerRules

PREDICATE EXAMPLE

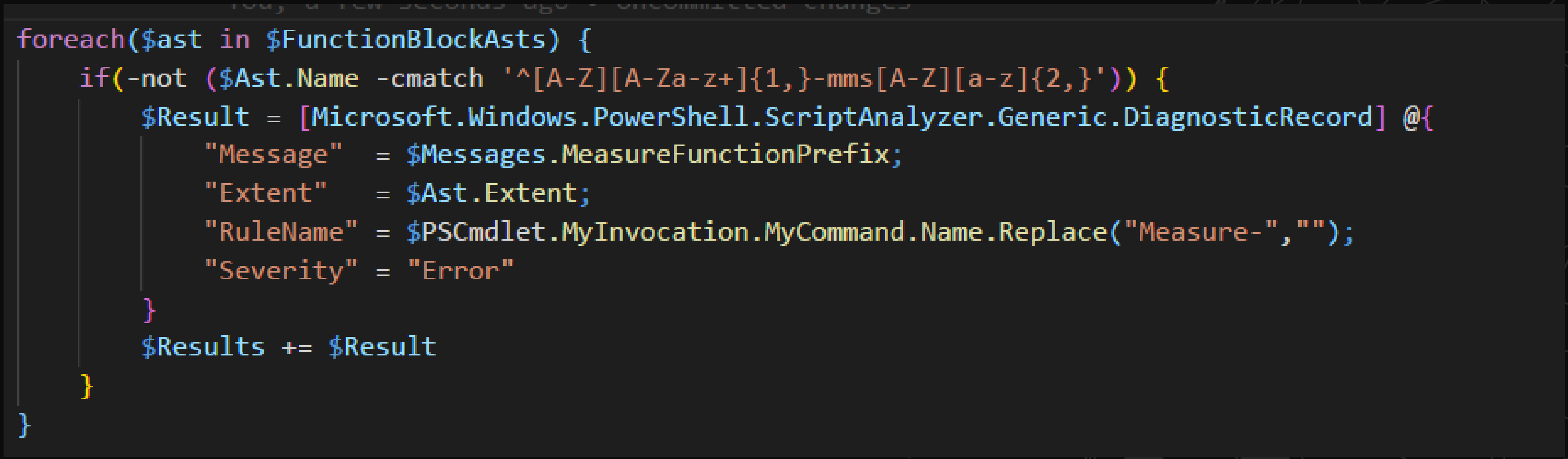


```
#Filter for just Functions
[ScriptBlock]$FunctionPredicate = {
    param (
        [System.Management.Automation.Language.Ast]$ast
    )
    [bool]$ReturnValue = $false

    If($ast -is [System.Management.Automation.Language.FunctionDefinitionAst]) {
        $ReturnValue = $true
    }
    return $ReturnValue
}
#Use the AST filter to find the functions in the scriptblock
[System.Management.Automation.Language.Ast[]]$FunctionBlockAsts
$FunctionBlockAsts = $ScriptBlockAst.FindAll($FunctionPredicate,$true)
```

MMS

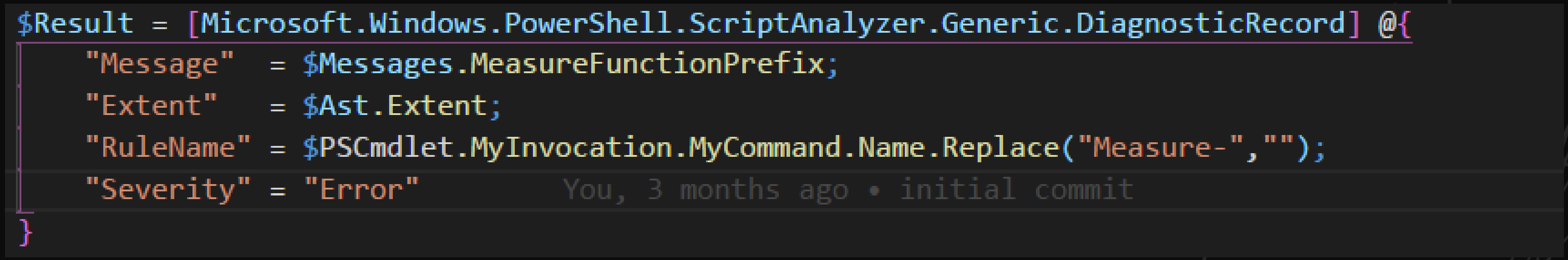
TEST YOUR RESULTS



```
foreach($ast in $FunctionBlockAsts) {
    if(-not ($Ast.Name -cmatch '^[A-Z][A-Za-z+]{1,}-mms[A-Z][a-z]{2,}')) {
        $Result = [Microsoft.Windows.PowerShell.ScriptAnalyzer.Generic.DiagnosticRecord] @{
            "Message"  = $Messages.MeasureFunctionPrefix;
            "Extent"   = $Ast.Extent;
            "RuleName" = $PSCmdlet.MyInvocation.MyCommand.Name.Replace("Measure-","");
            "Severity" = "Error"
        }
        $Results += $Result
    }
}
```

DIAGNOSTICRECORD

- This is splatting parameters to the Diagnostic record constructor.



```
$Result = [Microsoft.Windows.PowerShell.ScriptAnalyzer.Generic.DiagnosticRecord] @{
    "Message"  = $Messages.MeasureFunctionPrefix;
    "Extent"   = $Ast.Extent;
    "RuleName" = $PSCmdlet.MyInvocation.MyCommand.Name.Replace("Measure-","");
    "Severity" = "Error"         You, 3 months ago • initial commit
}
```

- Message - this is just a string that is displayed to the user. We wrapped it up using a little helper to make the code prettier. It is not needed. "Message" = "you wrote bad code here" would work too.
- Extent: This is the line number and some other meta data about your AST in relation to the bigger script.
- Rule Name: This is the rule name returned. (displayed to user)
- Severity: Error, Warning, Info. This must be exact.

MMS

Testing for Tokens!

$object | Slow-DownMyScript

https://github.com/fredbainbridge/CustomPSScriptAnalyzerRules

MMS

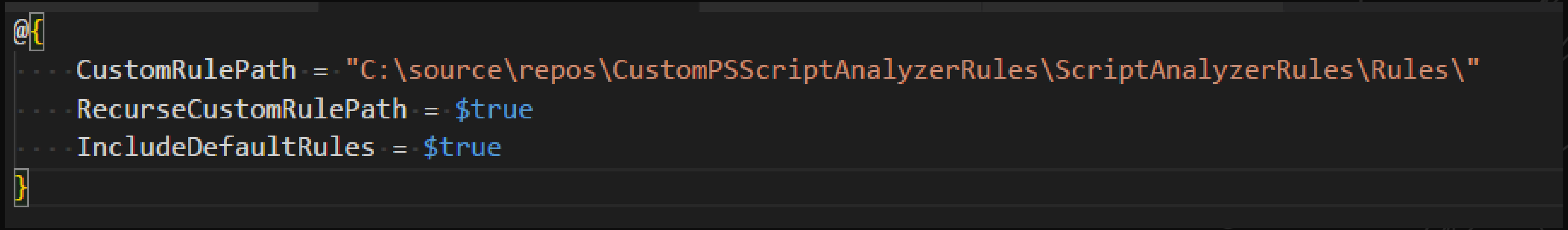
Integrating custom rules with VS Code

https://github.com/fredbainbridge/CustomPSScriptAnalyzerRules

MMS

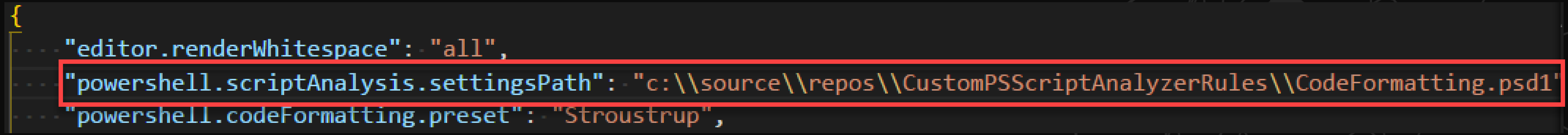
INTEGRATING WITH VS CODE

- 1. Create a CodeFormatting.psd1



```
@{
    CustomRulePath = "C:\source\repos\CustomPSScriptAnalyzerRules\ScriptAnalyzerRules\Rules\"
    RecurseCustomRulePath = $true
    IncludeDefaultRules = $true
}
```

- 2. Configure the PowerShell extension to use this code formatting definition.



```
{
    "editor.renderWhitespace": "all",
    "powershell.scriptAnalysis.settingsPath": "c:\\source\\repos\\CustomPSScriptAnalyzerRules\\CodeFormatting.psd1"
    "powershell.codeFormatting.preset": "Stroustrup",
```

Extended Q&A

MMS

RADISSON BLU · MALL OF AMERICA

MMS