PSCalendar Manual v2.9.0

Table of Contents

Introduction

This manual is a PDF version of several module-related reference files as well as all of the command help. The goal is to provide a single source for all module documentation. Be aware that many of the source files contain internal cross-references. Best efforts have been made to port those links to this document. External links should work as expected.

If you need to ask a question or report a problem, please visit the module's Github repository.

PSCalendar Overview

This module contains a few functions for displaying a calendar in the PowerShell console. The primary function is based on code originally published by Lee Holmes at http://www.leeholmes.com/blog/2008/12/03/showing-calendars-in-your-oof-messages/. However, `v2.0.0` of this module contains a complete rewrite of the core functions.

After installing the module, you can view a local PDF version of this file by running `Show-PSCalendarHelp`.

Installation

You can install this module from the PowerShell Gallery.

```
Install-Module PSCalendar [-scope currentuser]
```

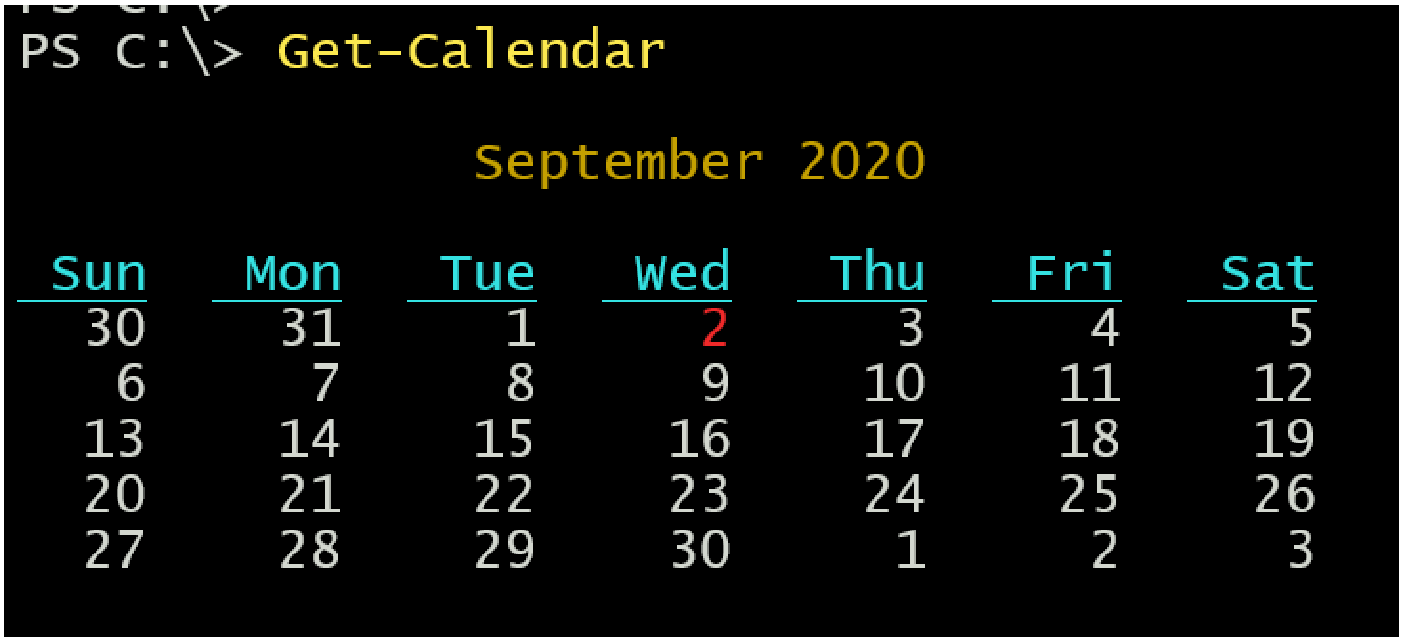
> Installing this module will also install the `ThreadJob` module from the PowerShell Gallery, as that is module dependency if you want to use `Show-GuiCalendar`.

The commands in this module have been tested on PowerShell 7 both under Windows and Linux and there is no reason these commands should not work. Commands and aliases that are incompatible with non-Windows platforms are not exported.

Note: If you are upgrading to v2.0.0 or later of this module, and have older versions installed, it is recommended that you uninstall the older versions.

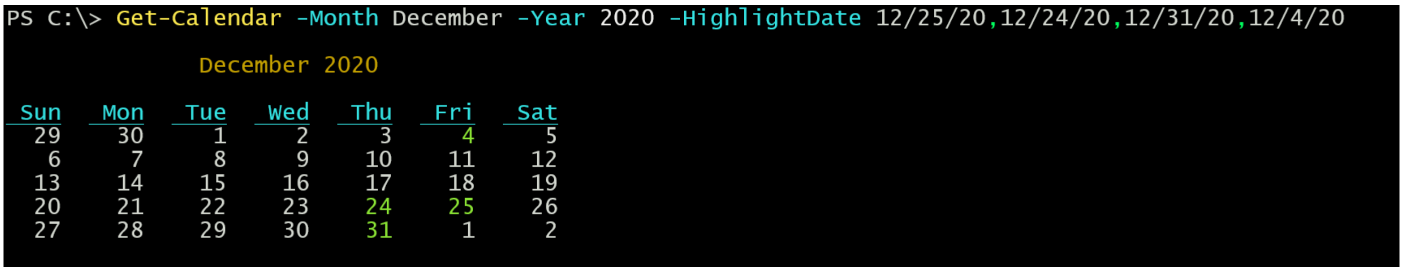
Get-Calendar

The commands in this module have been updated to take advantage ANSI escape sequences. The main function, Get-Calendar, will display the current month in the console, highlighting the current date with an ANSI escape sequence.



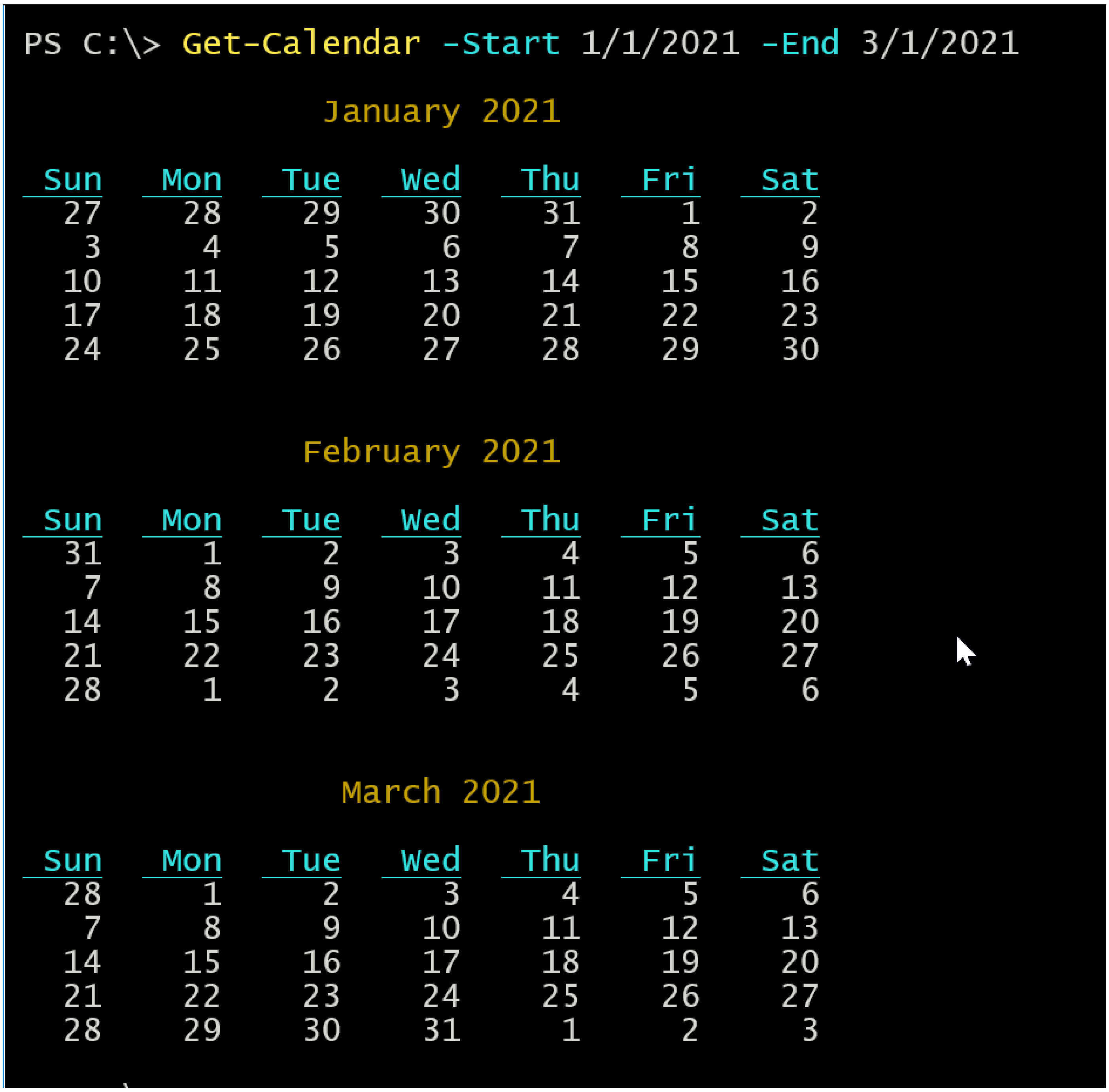
```
PS C:\> Get-Calendar

              September 2020

  Sun    Mon    Tue    Wed    Thu    Fri    Sat
   30     31      1      2      3      4      5
    6      7      8      9     10     11     12
   13     14     15     16     17     18     19
   20     21     22     23     24     25     26
   27     28     29     30      1      2      3
```

But you can also specify a calendar by month and year.



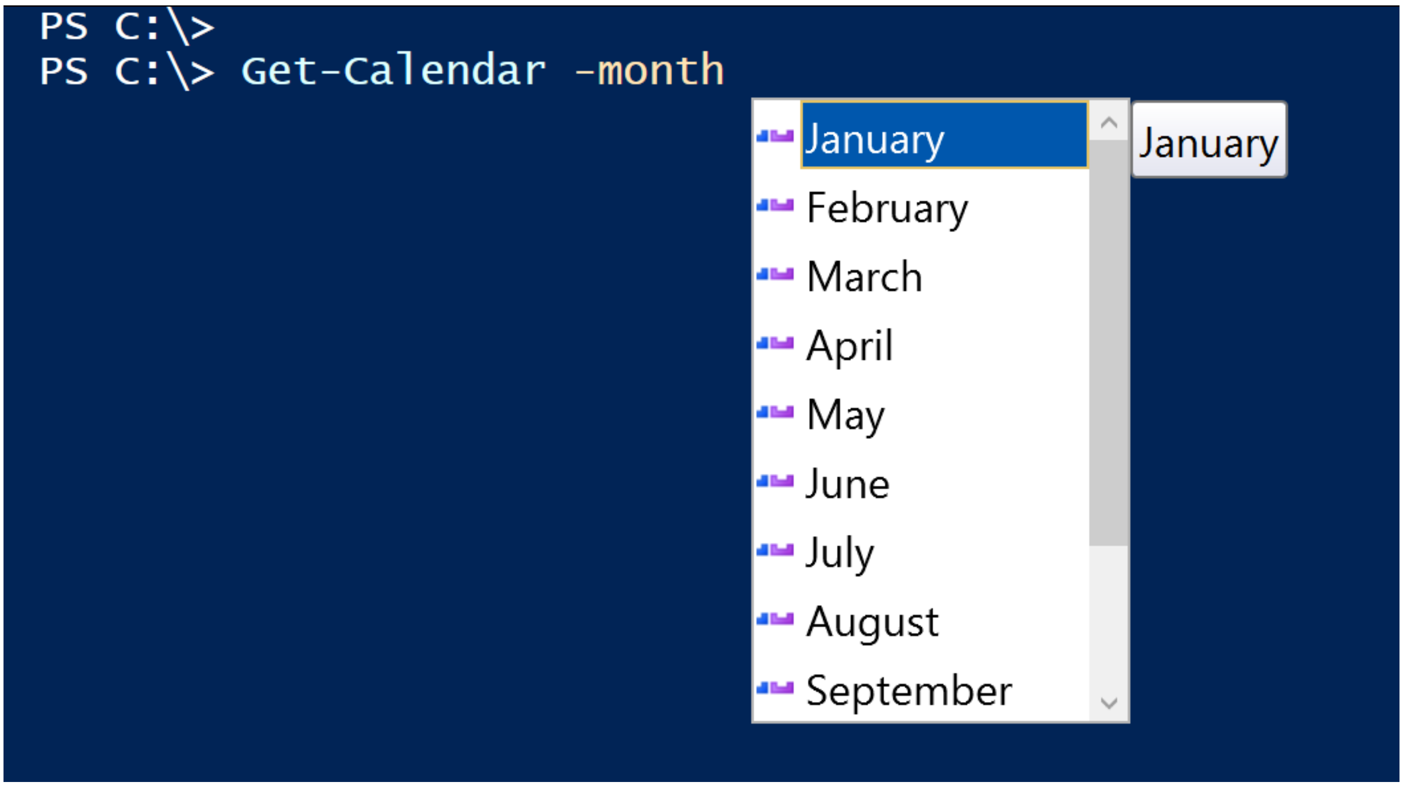
```
PS C:\> Get-Calendar -Month December -Year 2020 -HighlightDate 12/25/20,12/24/20,12/31/20,12/4/20

              December 2020

 Sun  Mon  Tue  Wed  Thu  Fri  Sat
  29   30    1    2    3    4    5
   6    7    8    9   10   11   12
  13   14   15   16   17   18   19
  20   21   22   23   24   25   26
  27   28   29   30   31    1    2
```

In this example you can see that I specified dates to highlight. Or you can specify a range of months.



```
PS C:\> Get-Calendar -Start 1/1/2021 -End 3/1/2021

               January 2021

  Sun    Mon    Tue    Wed    Thu    Fri    Sat
   27     28     29     30     31      1      2
    3      4      5      6      7      8      9
   10     11     12     13     14     15     16
   17     18     19     20     21     22     23
   24     25     26     27     28     29     30


              February 2021

  Sun    Mon    Tue    Wed    Thu    Fri    Sat
   31      1      2      3      4      5      6
    7      8      9     10     11     12     13
   14     15     16     17     18     19     20
   21     22     23     24     25     26     27
   28      1      2      3      4      5      6


                March 2021

  Sun    Mon    Tue    Wed    Thu    Fri    Sat
   28      1      2      3      4      5      6
    7      8      9     10     11     12     13
   14     15     16     17     18     19     20
   21     22     23     24     25     26     27
   28     29     30     31      1      2      3
```

The function should be culturally aware. The commands in this module that have a `-Month` parameter should autocomplete to culture-specific month names.

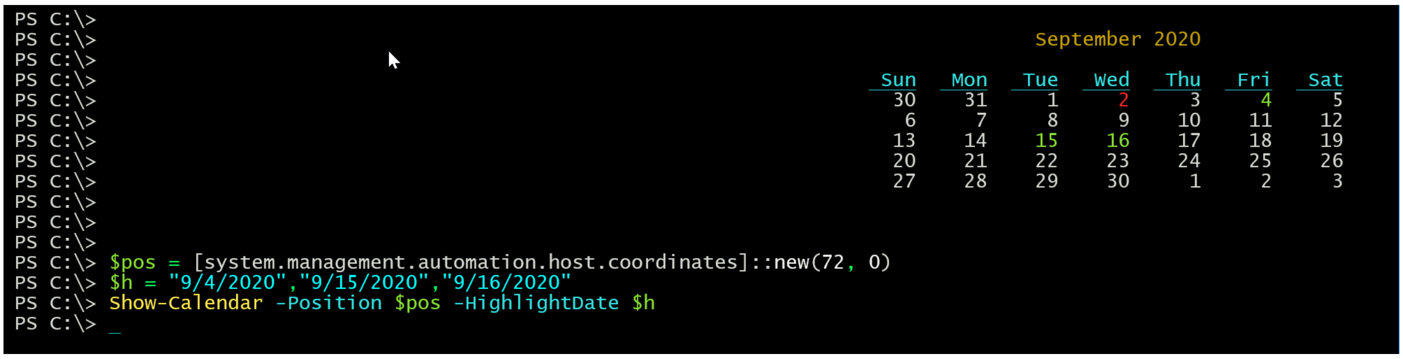


```
PS C:\>
PS C:\> Get-Calendar -month
```

There is a similar autocompletion for `-Year` that begins with the current year and then the next 5 years. Although nothing prevents you from entering any year you want.

Show-Calendar

In previous versions of this module, there was a command called `Show-Calendar` which wrote a colorized version of the calendar to the host using `Write-Host`. This command has been rewritten and now is essentially a wrapper for `Get-Calendar`. The primary difference is that you can position the calendar.



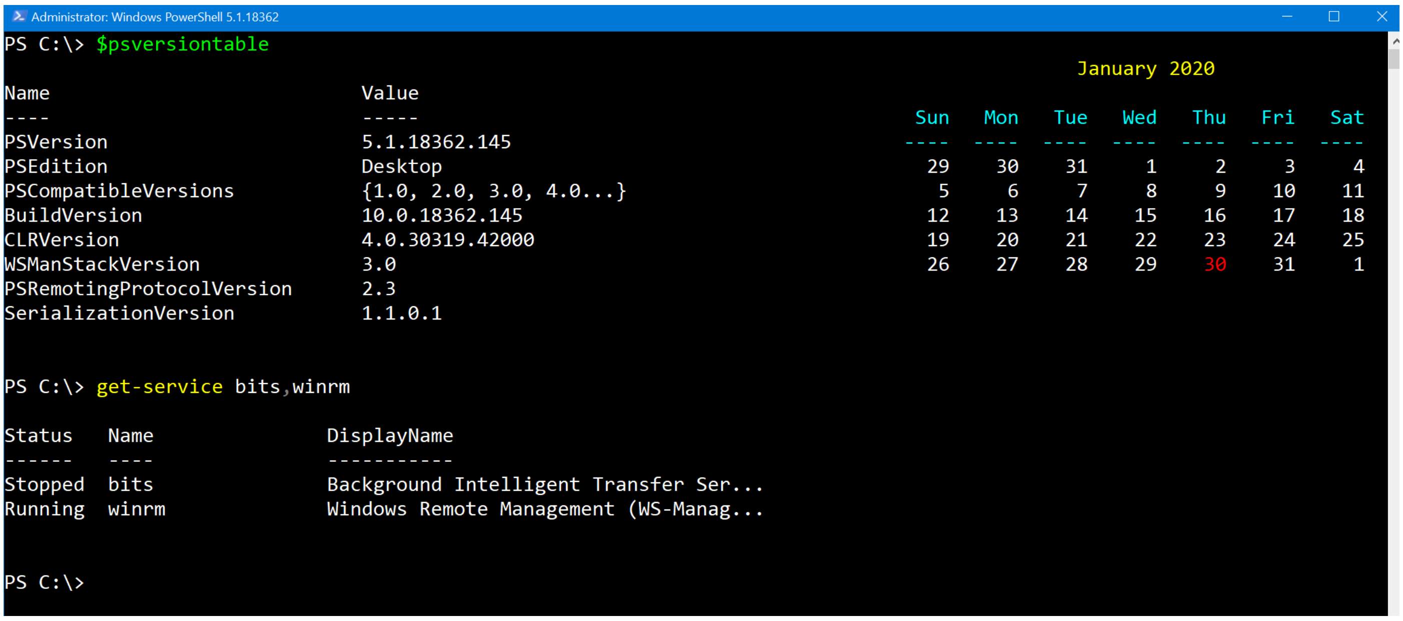
```
PS C:\>
PS C:\>                                                                       September 2020
PS C:\>
PS C:\>                                                              Sun    Mon    Tue    Wed    Thu    Fri    Sat
PS C:\>                                                               30     31      1      2      3      4      5
PS C:\>                                                                6      7      8      9     10     11     12
PS C:\>                                                               13     14     15     16     17     18     19
PS C:\>                                                               20     21     22     23     24     25     26
PS C:\>                                                               27     28     29     30      1      2      3
PS C:\>
PS C:\>
PS C:\>
PS C:\> $pos = [system.management.automation.host.coordinates]::new(72, 0)
PS C:\> $h = "9/4/2020","9/15/2020","9/16/2020"
PS C:\> Show-Calendar -Position $pos -HighlightDate $h
PS C:\> _
```

A Console Calendar Prompt

One way you might want to use this is in your PowerShell console. You can use the prompt function like this:

```
#requires -modules @{ModuleName="PSCalendar";ModuleVersion="2.5.0"}

Function prompt {

  #define a buffercell fill
  $fill = [system.management.automation.host.buffercell]::new(" ",$host.ui.RawUI.BackgroundColor,$host.ui.RawUI.BackgroundColor,"complete")

  #define a rectangle with an upper left corner X distance from the edge
  $left =$host.ui.RawUI.WindowSize.width - 42

  #need to adjust positioning based on buffer size of the console
  #is the cursor beyond the window size, ie have we scrolled down?
    if ($host.UI.RawUI.CursorPosition.Y -gt $host.UI.RawUI.WindowSize.Height) {
        $top = $host.ui.RawUI.CursorPosition.Y - $host.UI.RawUI.WindowSize.Height
    }
    else {
        $top = 0
    }
  #    System.Management.Automation.Host.Rectangle new(int left, int top, int right, int bottom)
  $r = [System.Management.Automation.Host.Rectangle]::new($left, 0, $host.ui.rawui.windowsize.width,$top+10)

  #clear the area for the calendar display
  $host.ui.rawui.SetBufferContents($r,$fill)

  #show the calendar in the upper right corner of the console
  $pos = [system.management.automation.host.coordinates]::new($left,0)
  Show-Calendar -Position $pos

  "PS $($executionContext.SessionState.Path.CurrentLocation)$('>' * ($nestedPromptLevel + 1)) ";

# .Link
# https://go.microsoft.com/fwlink/?LinkID=225750
# .ExternalHelp System.Management.Automation.dll-help.xml

}
```

Assuming the width of your console is at least 120, this code should work. Otherwise, you might need to tweak the positioning. This should also work in Windows Terminal. If you add some highlighted dates using `$PSDefaultParameterValues`, then you'll have a calendar right in front of you.



```
PS C:\> $psversiontable
                                                                       January 2020
Name                           Value
----                           -----                              Sun   Mon   Tue   Wed   Thu   Fri   Sat
PSVersion                      5.1.18362.145                      ----  ----  ----  ----  ----  ----  ----
PSEdition                      Desktop                              29    30    31     1     2     3     4
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0...}               5     6     7     8     9    10    11
BuildVersion                   10.0.18362.145                       12    13    14    15    16    17    18
CLRVersion                     4.0.30319.42000                      19    20    21    22    23    24    25
WSManStackVersion              3.0                                  26    27    28    29    30    31     1
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1


PS C:\> get-service bits,winrm

Status   Name               DisplayName
------   ----               -----------
Stopped  bits               Background Intelligent Transfer Ser...
Running  winrm              Windows Remote Management (WS-Manag...


PS C:\>
```

Note that any command output may be truncated because of the calendar display. This prompt function works as expected when using the Windows Terminal. Function needs work to behave as expected in a traditional

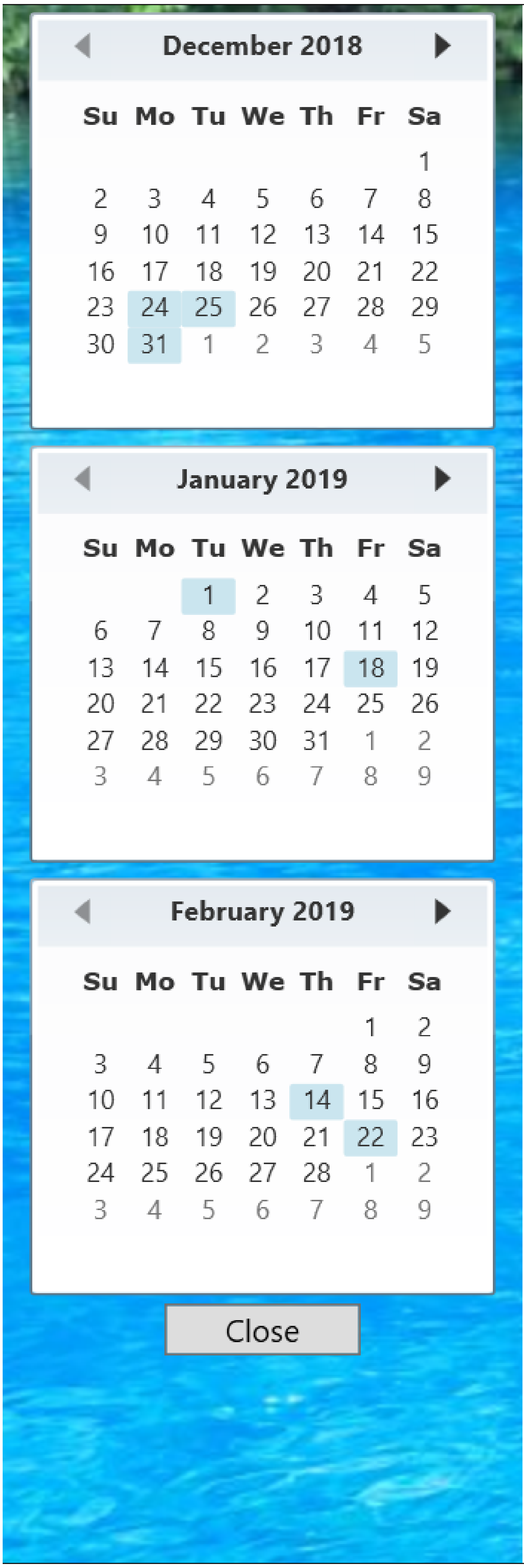
PowerShell console where you might have a large buffer for scrolling.

Show-GUICalendar

Finally, you can display a graphical calendar using a Windows Presentation Foundation (WPF) based script.

The function runs the calendar-related code in a runspace so it does not block your prompt. You can display up to 3 months and specify dates to highlight.

```
PS C:\> Show-GuiCalendar 12/2018 2/2019 -highlight 12/24/18,12/25/18,12/31/18,1/1/19,1/18/19,2/14/19,2/22/19
```



December 2018

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

January 2019

Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

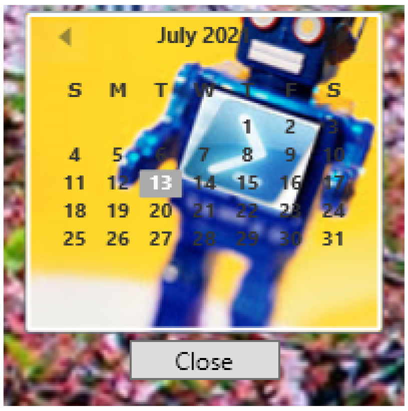
February 2019

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	1	2
3	4	5	6	7	8	9

The calendar form is transparent. But you should be able to click on it to drag it around your screen. You can also use the + and `-` keys to increase or decrease the calendar's opacity. Be aware that if you close the PowerShell session that launched the calendar, the calendar too will close.

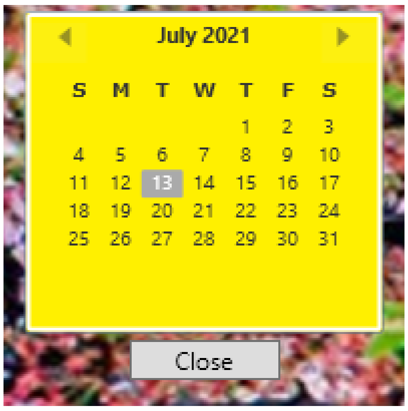
Beginning with module version 2.2.0 you can also customize the calendar background with an image:

```
Show-GuiCalendar -BackgroundImage D:\images\blue-robot-ps-thumb.jpg -Stretch UniformToFill -FontWeight Bold
```



Or you can specify a color. You can specify a WPF brush color like Cornsilk or Wheat, or use a color code like `#FFF000`:

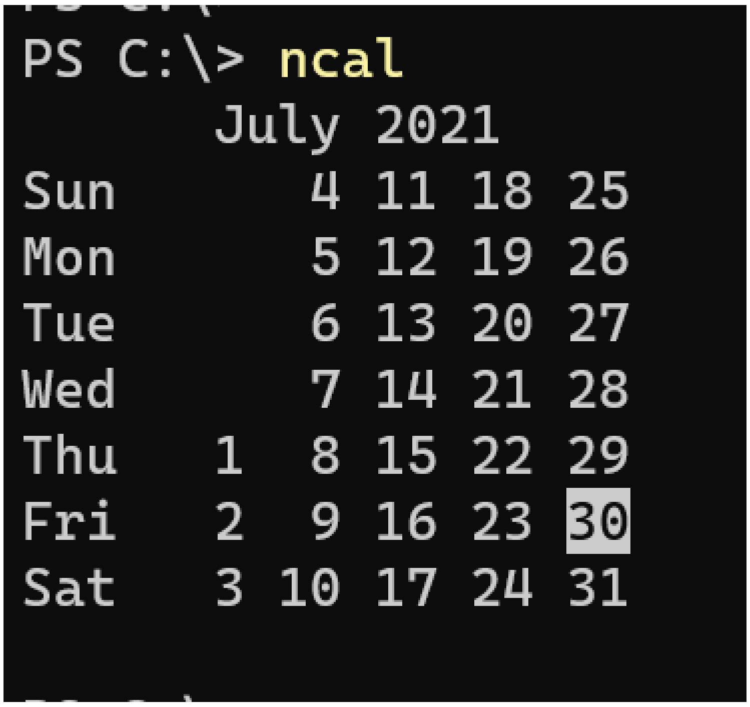
```
Show-GuiCalendar -BackgroundColor "#FFF000"
```



On Windows platforms, the `-BackgroundColor` parameter will autocomplete the available brush colors.

Get-NCalendar

The Linux world has an *ncal* command which displays the month in a vertical fashion. `Get-NCalendar` and its alias `ncal` work in a similar manner. The default is for the current month and year.



```
PS C:\> ncal
      July 2021
Sun        4 11 18 25
Mon        5 12 19 26
Tue        6 13 20 27
Wed        7 14 21 28
Thu     1  8 15 22 29
Fri     2  9 16 23 30
Sat     3 10 17 24 31
```

The current date will be highlighted unless you use `-HideHighlight`. You must use the full month name, although there is tab completion.

```
PS C:\> ncal January 2022
    January 2022
Sun   2  9 16 23 30
Mon   3 10 17 24 31
Tue      4 11 18 25
Wed      5 12 19 26
Thu      6 13 20 27
Fri      7 14 21 28
Sat   1  8 15 22 29
```

This command does not support date highlighting. See below.

Get-MonthName

This simple command will list the full month names for the current culture.

```
PS C:\> Get-MonthName
January
February
March
April
May
June
July
August
September
October
November
December
```

You might use this to build a larger `ncal` listing.

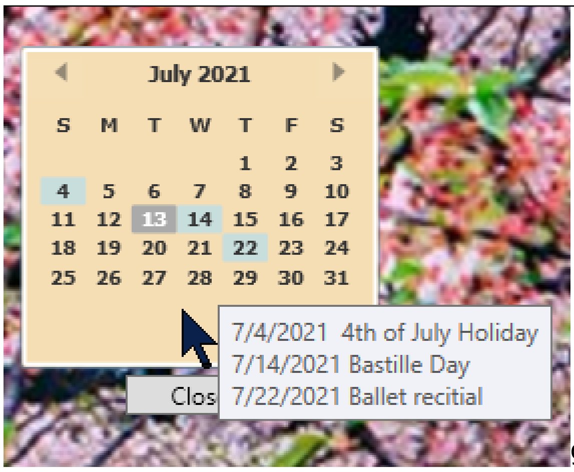
```
PS C:\> Get-MonthName | Select-Object -first 3 | Get-NCalendar -Year 2022
    January 2022
Sun   2  9 16 23 30
Mon   3 10 17 24 31
Tue      4 11 18 25
Wed      5 12 19 26
Thu      6 13 20 27
Fri      7 14 21 28
Sat   1  8 15 22 29


    February 2022
Sun      6 13 20 27
Mon      7 14 21 28
Tue      1  8 15 22
Wed      2  9 16 23
Thu      3 10 17 24
Fri      4 11 18 25
Sat      5 12 19 26


      March 2022
Sun      6 13 20 27
Mon      7 14 21 28
Tue   1  8 15 22 29
Wed   2  9 16 23 30
Thu   3 10 17 24 31
Fri      4 11 18 25
Sat      5 12 19 26
```

Highlight Dates with Notes

Beginning with v2.2.0, in addition to specifying an array of dates to highlight, you can also use a hashtable. The key should be the highlight date, and the value a brief description.

```
$h = @{"7/4/2021"="4th of July Holiday";"7/14/2021"="Bastille Day";"7/22/2021"="Ballet recitial"}
Show-GuiCalendar -Start 7/1/2021 -HighlightDate $h -BackgroundColor wheat -FontWeight Bold -Font Tahoma
```

When you pass a hashtable, you will get a tooltip popup when you hover the mouse over the month.



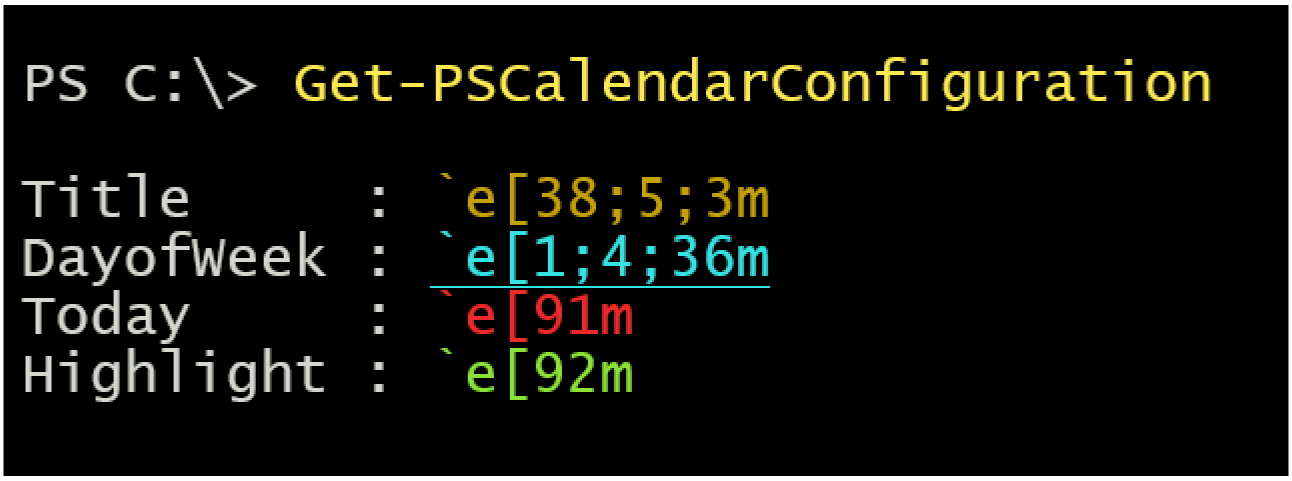
7/4/2021 4th of July Holiday
7/14/2021 Bastille Day
7/22/2021 Ballet recitial

g

This function requires the WPF-related assemblies. It should work in Windows PowerShell and PowerShell 7. You will receive a warning if any incompatibility is detected.

Customizing the Calendar Appearance

Beginning with v2.0.0 of this module, ANSI escape sequences used to format the calendar are stored in module-scoped hashtable. You can use Get-PSCalendarConfiguration to view the current settings.

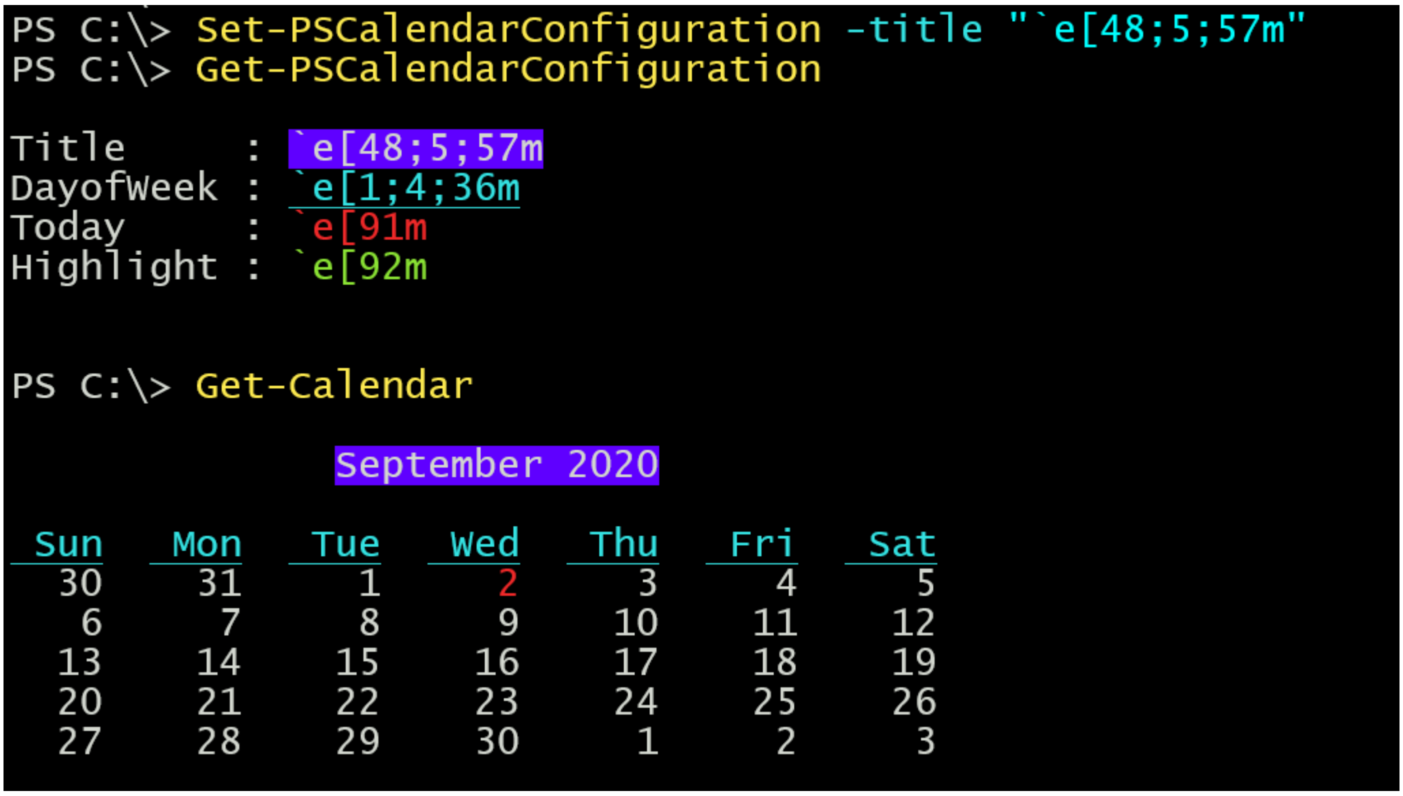


```
PS C:\> Get-PSCalendarConfiguration

Title      : `e[38;5;3m
DayofWeek  : `e[1;4;36m
Today      : `e[91m
Highlight  : `e[92m
```

The output will show you the escape sequence appropriate for your PowerShell version. If you want to change a setting, you can use:

Set-PSCalendarConfiguration

You need to include the escape character but you do not need to include the closing escape sequence.



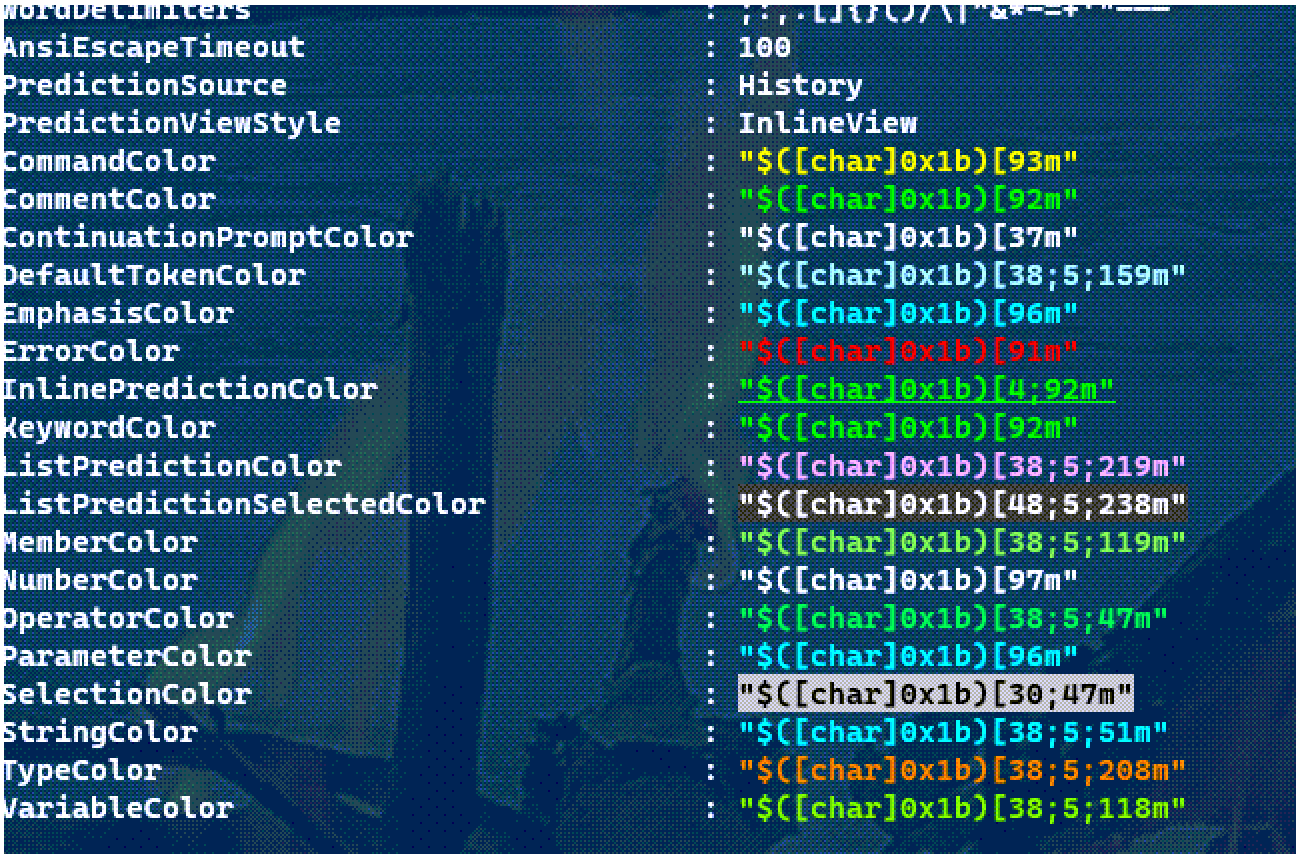
```
PS C:\> Set-PSCalendarConfiguration -title "`e[48;5;57m"
PS C:\> Get-PSCalendarConfiguration

Title     : `e[48;5;57m
DayofWeek : `e[1;4;36m
Today     : `e[91m
Highlight : `e[92m


PS C:\> Get-Calendar

                September 2020

  Sun    Mon    Tue    Wed    Thu    Fri    Sat
   30     31      1      2      3      4      5
    6      7      8      9     10     11     12
   13     14     15     16     17     18     19
   20     21     22     23     24     25     26
   27     28     29     30      1      2      3
```

This change lasts for the duration of your PowerShell session. If you want to make it more permanent, you will need to add the commands to your PowerShell profile script.

ANSI Support

As you’ve seen, there are several commands in this module that rely on ANSI for formatting. The hosting application needs to be able to recognize and use ANSI escape sequences. These commands should work in both Windows PowerShell and PowerShell 7 in the traditional PowerShell consoles or in Windows Terminal. They will not work in the PowerShell ISE. ANSI-related output will be automatically disabled if the PowerShell ISE is detected.

If you aren’t sure if your host supports ANSI, run `Get-PSReadlineOption`. You should see something like this:



```
WordDelimiters                      : ;:,.[]{}()/\|^&*-=+'"–—―
AnsiEscapeTimeout                   : 100
PredictionSource                    : History
PredictionViewStyle                 : InlineView
CommandColor                        : "$([char]0x1b)[93m"
CommentColor                        : "$([char]0x1b)[92m"
ContinuationPromptColor             : "$([char]0x1b)[37m"
DefaultTokenColor                   : "$([char]0x1b)[38;5;159m"
EmphasisColor                       : "$([char]0x1b)[96m"
ErrorColor                          : "$([char]0x1b)[91m"
InlinePredictionColor               : "$([char]0x1b)[4;92m"
KeywordColor                        : "$([char]0x1b)[92m"
ListPredictionColor                 : "$([char]0x1b)[38;5;219m"
ListPredictionSelectedColor         : "$([char]0x1b)[48;5;238m"
MemberColor                         : "$([char]0x1b)[38;5;119m"
NumberColor                         : "$([char]0x1b)[97m"
OperatorColor                       : "$([char]0x1b)[38;5;47m"
ParameterColor                      : "$([char]0x1b)[96m"
SelectionColor                      : "$([char]0x1b)[30;47m"
StringColor                         : "$([char]0x1b)[38;5;51m"
TypeColor                           : "$([char]0x1b)[38;5;208m"
VariableColor                       : "$([char]0x1b)[38;5;118m"
```

If you don’t see color formatting, the hosting application doesn’t support ANSI.

A Note on Culture

I've tried very hard to make the commands respect culture. Most commands now that string values to represent dates which are then treated as dates internally. For this reason, it is important that you follow the culture-specific short date format that you get from running this command:

```
(Get-Culture).datetimeformat.ShortDatePattern
```

In Windows PowerShell, all of the commands appear to respect culture settings. However, when running in PowerShell 7 there appears to be a bug in .NET Core and how it returns culture information for some cultures, specifically the first day of the week. If you run `Get-Calendar` or `Show-Calendar` and the week begins on the wrong day, use the `FirstDay` parameter to override the detected .NET values with the correct one.

```
PS C:\> Get-Calendar august -firstday Monday -highlight 1/8/2021,15,8,2021

               August 2021

 Mon    Tue    Wed    Thu    Fri    Sat    Sun
  26     27     28     29     30     31      1
   2      3      4      5      6      7      8
   9     10     11     12     13     14     15
  16     17     18     19     20     21     22
  23     24     25     26     27     28     29
  30     31      1      2      3      4
```

For example, if you are running under the en-AU culture, you would need to use this syntax.

Potential Issues

I have tried to make this module culture-aware. Testing across cultures is not an easy process. If you encounter a problem and are not running PowerShell under the `EN-US` culture, run the calendar command you are trying to use with `-Verbose` and post the results in a new issue. Or if you have both Windows PowerShell and PowerShell 7 installed, try the same command in both versions.

Module Commands

This section contains the same help content you would get from a PowerShell prompt using `Get-Help`. Note that most code examples have been formatted to fit the 80 character page width and sometimes with artificial formatting. Don't assume you can run examples *exactly* as they are shown. Some of the help examples might also use special or custom characters that might not render properly in the PDF.

Remember, you can also view the online help for each command:

```
Help Get-Calendar -online
```

If you can't remember what commands are in this module, you can always ask PowerShell.

```
Get-Command -module PSCalendar
```

Note that not all functions are available in PowerShell 7 on non- Windows platforms.

Get-Calendar

Synopsis

Displays a visual representation of a calendar.

Syntax

month (Default)

```
Get-Calendar [[-Month] <String>] [[-Year] <Int32>] [-HighlightDate <String[]>] [-FirstDay <DayOfWeek>] [-NoANSI] [-MonthOnly] [<CommonParameters>]
```

span

```
Get-Calendar -Start <String> -End <String> [-HighlightDate <String[]>] [-FirstDay <DayOfWeek>] [-NoANSI] [-MonthOnly] [<CommonParameters>]
```

calyear

```
Get-Calendar [-HighlightDate <String[]>] [-FirstDay <DayOfWeek>] [-NoANSI] [-MonthOnly] -CalendarYear <Int32> [<CommonParameters>]
```

Description

This command displays a visual representation of a calendar. It supports multiple months, as well as the ability to highlight a specific date or dates. The default display uses ANSI escape sequences. You can adjust the color scheme using Set-PSCalendarConfiguration.

When you enter Highlight, Start, or End dates, be sure to use the format that is culturally appropriate. It should match the pattern you get from running this command:

(Get-Culture).DateTimeFormat.ShortDatePattern

Examples

Example 1

```
PS C:\> Get-Calendar

                  July 2021

 Sun    Mon    Tue    Wed    Thu    Fri    Sat
  27     28     29     30      1      2      3
   4      5      6      7      8      9     10
  11     12     13     14     15     16     17
  18     19     20     21     22     23     24
  25     26     27     28     29     30     31
```

Show the current month. The current day will be formatted in color.

Example 2

```
PS C:\>  Get-Calendar -start "3/1/2021" -end "5/1/2021"
```

Display monthly calendars from March to May, 2021.

Example 3

```
PS C:\> Get-Calendar December -HighlightDate 12/4/2020,12/25/2020,12/24/2020,12/31/2020


              December 2020

 Sun    Mon    Tue    Wed    Thu    Fri    Sat
  29     30      1      2      3      4      5
   6      7      8      9     10     11     12
  13     14     15     16     17     18     19
  20     21     22     23     24     25     26
  27     28     29     30     31      1      2
```

Display a month and highlight specific dates in color.

Example 4

```
PS C:\> Get-Calendar august -firstday Monday -highlight 1/8/2021,15,8,2021

                August 2021

 Mon    Tue    Wed    Thu    Fri    Sat    Sun
  26     27     28     29     30     31      1
   2      3      4      5      6      7      8
   9     10     11     12     13     14     15
  16     17     18     19     20     21     22
  23     24     25     26     27     28     29
  30     31      1      2      3      4
```

In Windows PowerShell, all of the commands appear to respect culture settings. However, when running in PowerShell 7 there appears to be a bug in .NET Core and how it returns culture information for some cultures, specifically the first day of the week. If you run `Get-Calendar` or `Show-Calendar` and the week begins on the wrong day, use the `FirstDay` parameter to override the detected .NET values with the correct one. If you are running under the en-AU culture in PowerShell 7, you would need to run this command.

Example 5

```
PS C:\> Get-Calendar -NoANSI -Start 7/1/2021 -end 9/1/2021  | Out-File c:\work\Q3.txt
```

Get the calendars for a month of ranges with no ANSI formatting and save the output to a text file.

Example 6

```
PS C:\> Get-Calendar -Month January -Year 2022 -NoANSI -MonthOnly

              January 2022

 Sun   Mon   Tue   Wed   Thu   Fri   Sat
                                       1
   2     3     4     5     6     7     8
   9    10    11    12    13    14    15
  16    17    18    19    20    21    22
  23    24    25    26    27    28    29
  30    31
```

Suppress leading and trailing days from other months with the MonthOnly parameter.

Example 7

```
PS C:\> Get-Calendar -CalendarYear 2022 -NoANSI | Out-File c:\work\2022.txt
```

Create a yearly calendar for 2022 and save the output to a text file.

Parameters

-Month

Select a month to display. The command will default to the current year unless otherwise specified.

```
Type: String
Parameter Sets: month
Aliases:

Required: False
Position: 1
Default value: current month
Accept pipeline input: False
Accept wildcard characters: False
```

-Year

Select a year for the specified month.

```
Type: Int32
Parameter Sets: month
Aliases:

Required: False
Position: 2
Default value: Current year
Accept pipeline input: False
Accept wildcard characters: False
```

-Start

The first month to display. You must format the dates to match your culture. It should match the pattern you get from running this command:

(Get-Culture).DateTimeFormat.ShortDatePattern

```
Type: String
Parameter Sets: span
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-End

The last month to display. You must format the dates to match your culture. It should match the pattern you get from running this command:

(Get-Culture).DateTimeFormat.ShortDatePattern

```
Type: String
Parameter Sets: span
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-HighlightDate

Specific days (named) to highlight. These dates are color formatted using ANSI escape sequences. You must format the dates to match your culture. It should match the pattern you get from running this command:

(Get-Culture).DateTimeFormat.ShortDatePattern

```
Type: String[]
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: (Get-Date).date.toString()
Accept pipeline input: False
Accept wildcard characters: False
```

-FirstDay

Specify the first day of the week. There is a potential bug in .NET Core where the detected first day of the week is incorrect. If that is true for your culture, use this parameter to manually specify the correct first day of the week.

```
Type: DayOfWeek
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: ([System.Globalization.CultureInfo]::CurrentCulture).DateTimeFormat.FirstDayOfWeek
Accept pipeline input: False
Accept wildcard characters: False
```

-NoANSI

Do not use any ANSI formatting. The output will be plain-text. This also means that the current day and highlight dates will not be reflected in the output. This parameter has no affect when running the command in the PowerShell ISE. There is no color formatting when using this host.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-MonthOnly

Do not show any leading or trailing days from other months.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-CalendarYear

Enter a year between 1000 and 3000 to display in calendar view.

```
Type: Int32
Parameter Sets: calyear
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

Outputs

System.String

Notes

This command should have an alias of cal. This function was originally inspired from work by Lee Holmes at http://www.leeholmes.com/blog/2008/12/03/showing-calendars-in-your-oof-messages/.

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-Date

Set-PSCalendarConfiguration

Show-Calendar

Show-GuiCalendar

Get-NCalendar

Get-MonthName

Synopsis

Get the list of month names.

Syntax

```
Get-MonthName [-Short] [<CommonParameters>]
```

Description

This command will return a list of month names according to your current culture.

Examples

Example 1

```
PS C:\> Get-MonthName
January
February
March
April
May
June
July
August
September
October
November
December
```

Get the standard list of month names.

Example 2

```
PS C:\> Get-MonthName -Short
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
```

Get the list of short month names.

Parameters

-Short

Get short month names.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

String

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-Culture

Get-NCalendar

Synopsis

Display a Linux-style ncal calendar.

Syntax

```
Get-NCalendar [[-Month] <String>] [[-Year] <Int32>] [-HideHighlight] [-Monday] [<CommonParameters>]
```

Description

This command generates equivalent output to the ncal Linux command. This is not a 100% port of that command but it should provide similar results for the same month and year. You should enter the complete month name. There should be tab-completion for the month and year values. This command has an alias of ncal on Windows platforms.

If you run this command in the PowerShell ISE, there will be no highlighting or ANSI formatting.

Examples

Example 1

```
PS C:\> Get-NCalendar
     July 2021
Sun     4 11 18 25
Mon     5 12 19 26
Tue     6 13 20 27
Wed     7 14 21 28
Thu  1  8 15 22 29
Fri  2  9 16 23 30
Sat  3 10 17 24 31
```

Get the calendar for the current month. The current day will be highlighted.

Example 2

```
PS C:\> ncal April 2022
     April 2022
Sun     3 10 17 24
Mon     4 11 18 25
Tue     5 12 19 26
Wed     6 13 20 27
Thu     7 14 21 28
Fri  1  8 15 22 29
Sat  2  9 16 23 30
```

Get an ncal for April 2022. This example is using the ncal alias.

Example 3

```
PS C:\> Get-MonthName | ncal

    January 2021
Sun  3 10 17 24 31
Mon     4 11 18 25
Tue     5 12 19 26
Wed     6 13 20 27
Thu     7 14 21 28
Fri  1  8 15 22 29
Sat  2  9 16 23 30

   February 2021
Sun     7 14 21 28
Mon     1  8 15 22
Tue     2  9 16 23
Wed     3 10 17 24
Thu     4 11 18 25
Fri     5 12 19 26
Sat     6 13 20 27
...
```

Display an ncal listing for the entire year.

Parameters

-Month

Enter the full month name. You should be able to tab-complete the parameter value. The default is the current month.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: 0
Default value: current month
Accept pipeline input: True (ByPropertyName, ByValue)
Accept wildcard characters: False
```

-Year

Enter the 4 digit year. You should be able to tab-complete the next 5 years but you can enter any 4 digit year. The default is the current year.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: 1
Default value: current year
Accept pipeline input: True (ByPropertyName)
Accept wildcard characters: False
```

-HideHighlight

Don’t highlight the current date. This parameter is automatically set to true when running in the PowerShell ISE.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Monday

Start the week on Monday.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

String

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-Calendar

Get-MonthName

Get-PSCalendarConfiguration

Synopsis

Get the current PSCalendar ANSI configuration.

Syntax

```
Get-PSCalendarConfiguration [<CommonParameters>]
```

Description

Get-Calendar and Show-Calendar commands rely on ANSI escape sequences to colorize the output. You can use this command to view the current settings. The settings will use an appropriate escape character based on your PowerShell version. Use Set-PSCalendarConfiguration to modify the settings.

Examples

Example 1

```
PS C:\> Get-PSCalendarConfiguration

Title     : `e[38;5;3m
DayOfWeek : `e[1;4;36m
Today     : `e[91m
Highlight : `e[92m
```

The display will be formatted with the corresponding ANSI escape sequence. The escape character, will reflect your current PowerShell version.

Parameters

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

PSCalendarConfiguration

Notes

Related Links

Set-PSCalendarConfiguration

Set-PSCalendarConfiguration

Synopsis

Modify the PSCalendar ANSI configuration.

Syntax

```
Set-PSCalendarConfiguration [[-Title] <String>] [[-DayOfWeek] <String>]
[[-Today] <String>] [[-Highlight] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

Description

The Get-Calendar and Show-Calendar commands use ANSI escape sequences to colorize the output. Get-PSCalendarConfiguration will display the current settings. You can use Set-PSCalendarConfiguration to modify the settings. You will need to include the escape sequence appropriate for your PowerShell version. You do not need to include the closing escape sequence. If you are looking for suggestions, take a look at Get-PSReadlineOption.

Any configuration changes you make are only for the duration of your PowerShell session. If you want to make them more permanent, you can add Set-PSCalendarConfiguration commands to your PowerShell profile.

Examples

Example 1

```
PS C:\>  Set-PSCalendarConfiguration -title "$([char]27)[48;5;57m"
```

Change the title color scheme on Windows 5.1. Although, this would also work in PowerShell 7.

Example 2

```
PS C:\>  Set-PSCalendarConfiguration -DayOfWeek "`e[48;5;57m"
```

Change the week day headings color scheme on PowerShell 7.x.

Parameters

-Confirm

Prompts you for confirmation before running the cmdlet.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-DayOfWeek

Specify an ANSI sequence for the day of the week heading.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Highlight

Specify the ANSI escape sequence to highlight specified dates.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: 3
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Title

Specify the ANSI escape sequence for the calendar title which will be the month and year.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Today

Specify the ANSI escape sequence to highlight the current day.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: 2
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WhatIf

Shows what would happen if the cmdlet runs. The cmdlet is not run.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

None

Notes

Related Links

Get-PSCalendarConfiguration

Show-Calendar

Synopsis

Display a colorized calendar month in the console.

Syntax

month

```
Show-Calendar [[-Month] <String>] [[-Year] <Int32>] [-HighlightDate <String[]>] [-FirstDay <DayOfWeek>] [-Position <Coordinates>] [-MonthOnly]
[<CommonParameters>]
```

calyear

```
Show-Calendar [-HighlightDate <String[]>] [-FirstDay <DayOfWeek>] [-MonthOnly] -CalendarYear <Int32>  [<CommonParameters>]
```

Description

This command is a wrapper for Get-Calendar that essentially shows the same result. The only difference is that you can use Show-Calendar to display the calendar at a specific position in your PowerShell session. This function is also retained for backward compatibility.

Examples

Example 1

```
PS C:\> Show-Calendar
```

Display a colorized version of the current month.

Example 2

```
PS C:\> Show-Calendar -Month February -Year 2021 -HighlightDate 2/22/21
```

Display February 2021 and highlight the 22nd using the default highlight color.

Example 3

```
PS C:\> Show-Calendar  -Position ([system.management.automation.host.coordinates]::new(75,1))
```

Display the calendar at a specified X,Y position in the console. This parameter will probably not work in the PowerShell ISE.

Example 4

```
PS C:\> Show-Calendar -Month January -Year 2022 -MonthOnly

               January 2022

 Sun    Mon    Tue    Wed    Thu    Fri    Sat
                                             1
   2      3      4      5      6      7      8
   9     10     11     12     13     14     15
  16     17     18     19     20     21     22
  23     24     25     26     27     28     29
  30     31
```

Suppress leading and trailing days from other months with the MonthOnly parameter.

Parameters

-Month

Select a month to display. The command will default to the current year unless otherwise specified.

```
Type: String
Parameter Sets: month
Aliases:

Required: False
Position: 1
Default value: Current month
Accept pipeline input: False
Accept wildcard characters: False
```

-Year

Select a year for the specified month.

```
Type: Int32
Parameter Sets: month
Aliases:

Required: False
Position: 2
Default value: current year
Accept pipeline input: False
Accept wildcard characters: False
```

-HighlightDate

Specify days to highlight. These dates are colored by ANSI escape sequences. You can modify them with Set-PSCalendarConfiguration. You must format the dates tpo match your culture. It should match the pattern you get from running this command: (Get-Culture).DateTimeFormat.ShortDatePattern

```
Type: String[]
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Position

Enter a System.Management.Automation.Host.Coordinates object to specify a location for the calendar. This may not work properly in all hosts or PowerShell versions and you might need some trial and error to figure out a position that works for you.

```
Type: Coordinates
Parameter Sets: month
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-FirstDay

Specify the first day of the week. There is a potential bug in .NET Core where the detected first day of the week is incorrect. If that is true for your culture, use this parameter to manually specify the correct first day of the week.

```
Type: DayOfWeek
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: ([System.Globalization.CultureInfo]::CurrentCulture).DateTimeFormat.FirstDayOfWeek
Accept pipeline input: False
Accept wildcard characters: False
```

-MonthOnly

Do not show any leading or trailing days.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-CalendarYear

Enter a year between 1000 and 3000 to display in calendar view.

```
Type: Int32
Parameter Sets: calyear
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

System.String

Notes

This command should have an alias of scal.

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Get-Calendar

Show-GuiCalendar

Show-GuiCalendar

Synopsis

Display a WPF-based calendar.

Syntax

basic (Default)

```
Show-GuiCalendar [[-Start] <String>] [[-End] <String>]
[-HighlightDate <Object[]>] [-Font <String>] [-FontStyle <String>]
[-FontWeight <String>] [-FirstDay <DayOfWeek>] [<CommonParameters>]
```

bgimage

```
Show-GuiCalendar [[-Start] <String>] [[-End] <String>]
[-HighlightDate <Object[]>] [-Font <String>] [-FontStyle <String>]
[-FontWeight <String>] [-BackgroundImage <String>] [-Stretch <String>]
[-FirstDay <DayOfWeek>] [<CommonParameters>]
```

bgcolor

```
Show-GuiCalendar [[-Start] <String>] [[-End] <String>]
[-HighlightDate <Object[]>] [-Font <String>] [-FontStyle <String>]
[-FontWeight <String>] [-BackgroundColor <String>] [-FirstDay <DayOfWeek>]
[<CommonParameters>]
```

Description

If you are running Windows PowerShell or a version of PowerShell that supports the [System.Windows.Media] .NET class, which does NOT include PowerShell 7, you can display a graphical calendar. You can specify up to 3 months. There are also parameters to fine-tune the calendar style. The calendar form itself is transparent, but you should be able to click on it to drag it around your screen. You can also use the + and - keys to increase or decrease the calendar’s opacity. You may have to click on a calendar before making any adjustments.

This command launches the calendar in a separate runspace so that it doesn’t block your prompt. However, if you close the PowerShell session that launched the calendar, the calendar will also automatically close.

You must format the dates to match your culture. It should match the pattern you get from running this command:

(Get-Culture).datetimeformat.ShortDatePattern

Examples

Example 1

```
PS C:\> Show-GuiCalendar
```

Display the current month as a graphical calendar.

Example 2

```
PS C:\> Show-GuiCalendar -start 12/2020 -end 2/2021 -highlight 12/24/19,12/25/19,12/31/19,1/1/20,2/14/20 -font 'Century Gothic' -FontStyle italic
```

Display 3 months with selected dates highlighted and style the calendar to font settings.

Example 3

```
PS C:\> Show-GuiCalendar -fontweight bold -backgroundcolor Azure
```

Display the current month with a bold font and the specified background color.

Example 4

```
PS C:\> $h = @{"7/4/2021"="4th of July";"7/14/2021"="Bastille Day";"7/22/2021"="Family Visit"}
PS C:\> Show-GuiCalendar -start 7/1/2021 -backgroundimage c:\scripts\zazu.gif -highlightDate $h
```

Display July 2021 with a background image and use the hashtable of highlight dates. A highlight summary will be displayed as a tool tip for the month.

Parameters

-End

Enter the last month to display by date, like 3/1/2020. You cannot display more than 3 months.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: 2
Default value: Current month
Accept pipeline input: False
Accept wildcard characters: False
```

-Font

Select a font family for your calendar.

```
Type: String
Parameter Sets: (All)
Aliases:
Accepted values: Segoi UI, QuickType, Tahoma, Lucida Console, Century Gothic

Required: False
Position: Named
Default value: Segoi UI
Accept pipeline input: False
Accept wildcard characters: False
```

-FontStyle

Select a font style for your calendar.

```
Type: String
Parameter Sets: (All)
Aliases:
Accepted values: Normal, Italic, Oblique

Required: False
Position: Named
Default value: Normal
Accept pipeline input: False
Accept wildcard characters: False
```

-FontWeight

Select a font weight for your calendar.

```
Type: String
Parameter Sets: (All)
Aliases:
Accepted values: Normal, DemiBold, Light, Bold

Required: False
Position: Named
Default value: Normal
Accept pipeline input: False
Accept wildcard characters: False
```

-HighlightDate

Enter an array of dates to highlight like 12/25/202,12/31/2020 or a hashtable where the key is the date and the value is a description. This data will be displayed as a tooltip for each month.

```
Type: Object[]
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Start

Enter the first month to display by date, like 1/1/2020.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: 1
Default value: Current month
Accept pipeline input: False
Accept wildcard characters: False
```

-BackgroundColor

Specify calendar background color. On Windows platforms you should be able to tab complete possible colors. Or you can use a color format like '#FFF000'. Remember to wrap this kind of value in quotes.

```
Type: String
Parameter Sets: bgcolor
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-BackgroundImage

Specify the path to an image to use as the background.

```
Type: String
Parameter Sets: bgimage
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-FirstDay

Specify the first day of the week. Normally, .NET should detect the first day of the week based on culture settings. But if for some reason, .NET is detecting incorrect information, you can manually set the first day of the week with this parameter.

```
Type: DayOfWeek
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Stretch

Specify image stretch setting. Possible values are None, Fill, Uniform, and UniformToFill

```
Type: String
Parameter Sets: bgimage
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

None

Notes

This function requires the WPF-related assemblies. It should work in Windows PowerShell and PowerShell 7 on Windows. You will receive a warning if any incompatibility is detected.

This command should have an alias of gcal.

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

Show-Calendar

Get-Calendar

Show-PSCalendarHelp

Synopsis

Display a help PDF file for the PSCalendar module.

Syntax

```
Show-PSCalendarHelp [<CommonParameters>]
```

Description

This command will open a local PDF copy of the module’s README file. The file will contain additional information and screenshots. The command will open the file using whatever application you have associated with the .PDF file extension.

Examples

Example 1

```
PS C:\> Show-PSCalendarHelp
```

Parameters

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

Inputs

None

Outputs

None

Notes

Learn more about PowerShell: http://jdhitsolutions.com/blog/essential-powershell-resources/

Related Links

https://github.com/jdhitsolutions/PSCalendar