PSCalendar Manual v2.10.0

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

PSCalendar Overview

Installation

You can install this module from the PowerShell Gallery.

```
Install-PSResource PSCalendar
```

> Installing this module will also install the `Microsoft.PowerShell.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 will be handled on a per command basis.

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

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.
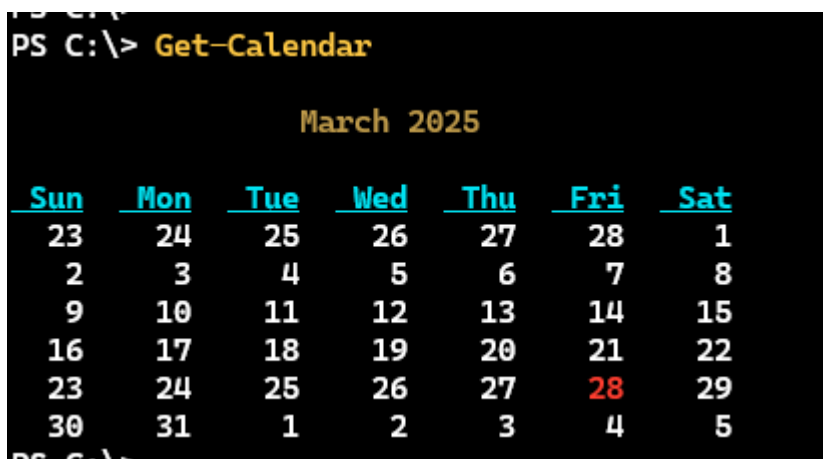
Module Commands

Name	Alias	Synopsis
Export-PSCalendarConfiguration	*Save-PSCalendarConfiguration*	Save the current calendar configuration settings to a file.
Get-Calendar	*cal*	Displays a visual representation of a calendar.
Get-MonthName	*mon*	Get the list of month names.
Get-NCalendar	*ncal*	Display a Linux-style ncal calendar.
Get-PSCalendarConfiguration		Get the current PSCalendar ANSI configuration.
Set-PSCalendarConfiguration		Modify the PSCalendar ANSI configuration.
Show-Calendar	*scal*	Display a colorized calendar month in the console.
Show-GuiCalendar	*gcal*	Display a WPF-based calendar.
Show-PSCalendarHelp		Display a help PDF file for the PSCalendar module.

Here are a few details are the commands you are most likely to use.

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.

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

```
PS C:\> Get-Calendar -Month December -Year 2025 -HighLightDate 12/25/25,12/24/25,12/31/25

              December 2025

 Sun   Mon   Tue   Wed   Thu   Fri   Sat
  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     3
```

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 1/1/2025 -end 3/1/2025

              January 2025

 Sun   Mon   Tue   Wed   Thu   Fri   Sat
  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


              February 2025

 Sun   Mon   Tue   Wed   Thu   Fri   Sat
  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     1


               March 2025

 Sun   Mon   Tue   Wed   Thu   Fri   Sat
  23    24    25    26    27    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     4     5
PS C:\>
```

In v2.10.0 you can now specify a range of months by calendar quarter. The default is the current year.

```
PS C:\> Get-Calendar -Quarter 3

                     July 2025

   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


                    August 2025

   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
    31      1      2      3      4      5      6

                  September 2025

   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     29     30      1      2      3      4
```
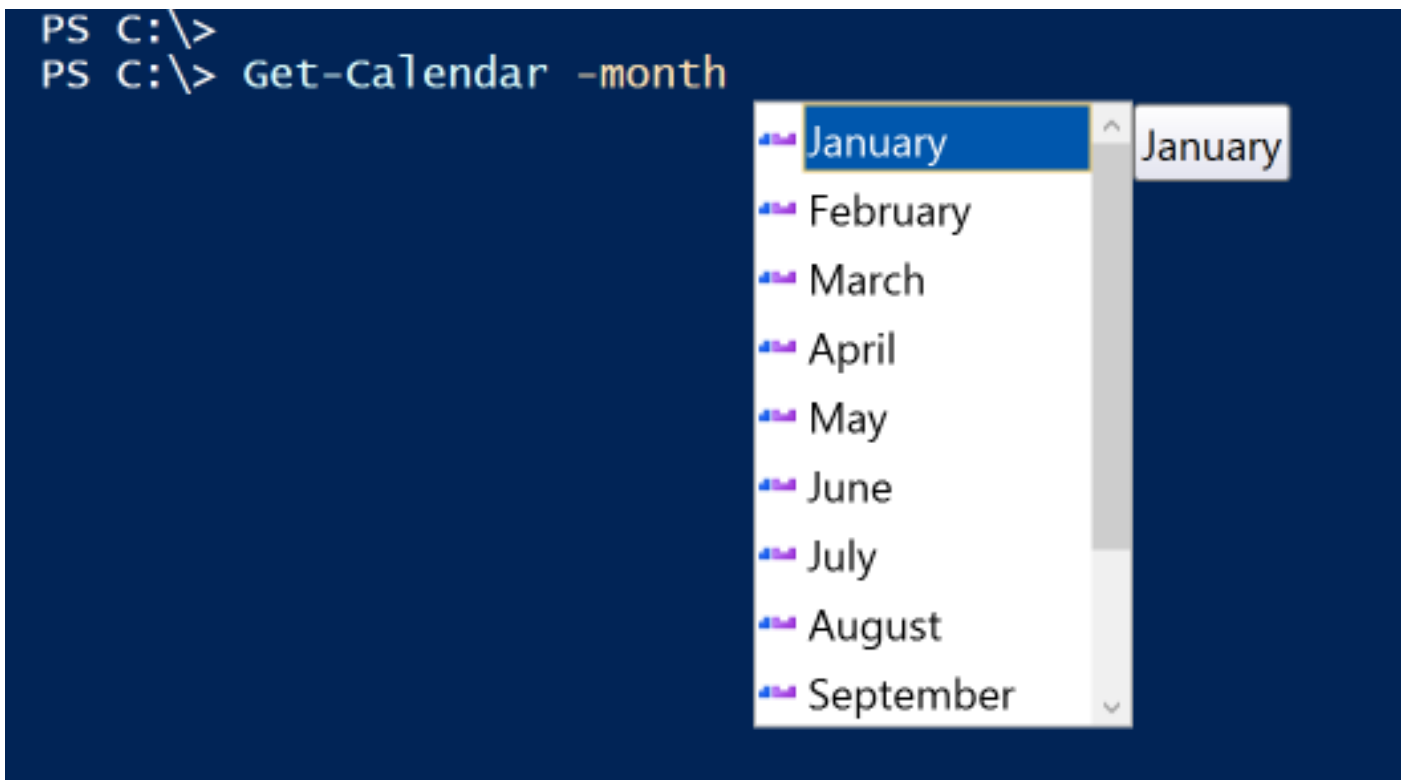
The function should be culturally aware. The commands in this module that have a -Month parameter should autocomplete to culture-specific month names.

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 with this command.

```
$pos = [System.Management.Automation.Host.Coordinates]::new(72,0)
$h = "7/4/2025","7/14/2025"
Show-Calendar -HighLightDate $h -Position $pos -Month July
```
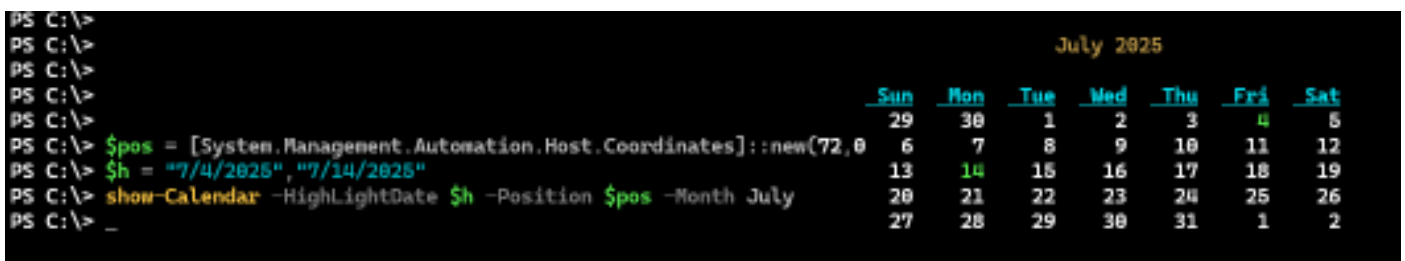

A Console Calendar Prompt

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

```
#requires -modules "PSCalendar"

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.

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.

> **This command is not supported on non-Windows platforms.**

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.

```
Show-GuiCalendar 12/2025 2/2026 -highlight 12/24/25,12/25/25,12/31/25,1/1/26,1/18/26,2/14/26,2/22/26
```

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-thumb3.png -Stretch Fill -FontWeight DemiBold
```


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"
```
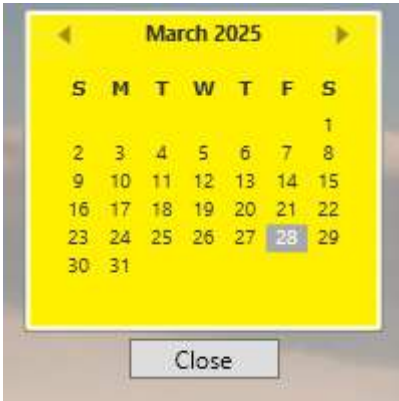

On Windows platforms, the `-BackgroundColor` parameter will autocomplete the available brush colors.

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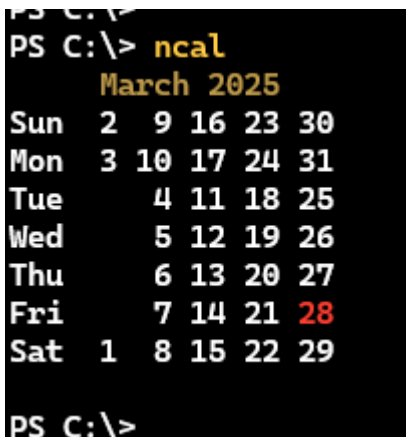
The current date will be highlighted unless you use `-HideHighlight`. You must use the full month name, although there is tab completion.

```
PS C:\> ncal July 2025
      July 2025
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
```

This command does not support date highlighting other than the current date. 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 2025
     January 2025
Sun      5 12 19 26
Mon      6 13 20 27
Tue      7 14 21 28
Wed  1   8 15 22 29
Thu  2   9 16 23 30
Fri  3 10 17 24 31
Sat      4 11 18 25

    February 2025
Sun      2  9 16 23
Mon      3 10 17 24
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

      March 2025
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
```

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 = @{"7/4/2025"="4th of July Holiday";"7/14/2025"="Bastille Day";"7/22/2025"="Ballet recital"}
Show-GuiCalendar -Start 7/1/2025 -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.
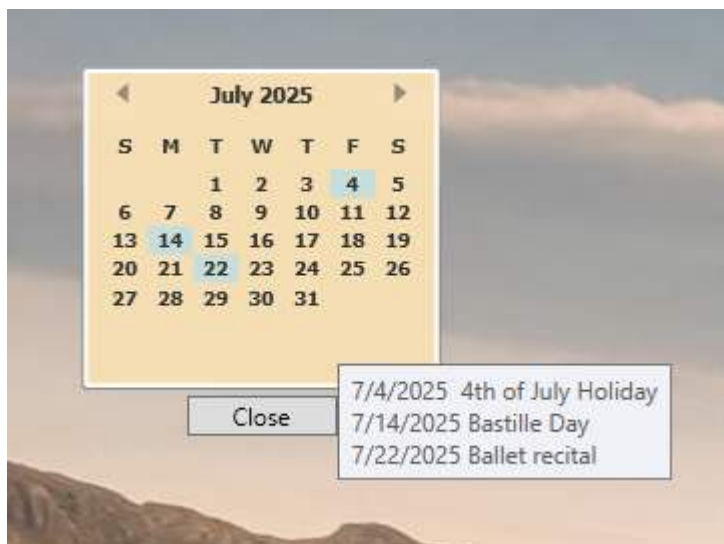
This function reqUIres the WPF-related assemblies. It should work in Windows PowerShell and PowerShell 7. You will receive a warning if any incompatibility is detected.

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.

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.

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.

Or you can use Export-PSCalendarConfiguration to save the settings to a file. This command was added in v2.10.0.

```
Export-PSCalendarConfiguration -passthru

        Directory: C:\Users\Jeff


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          3/29/2025   2:51 PM            136 ▯  .pscalendarConfiguration.json
```
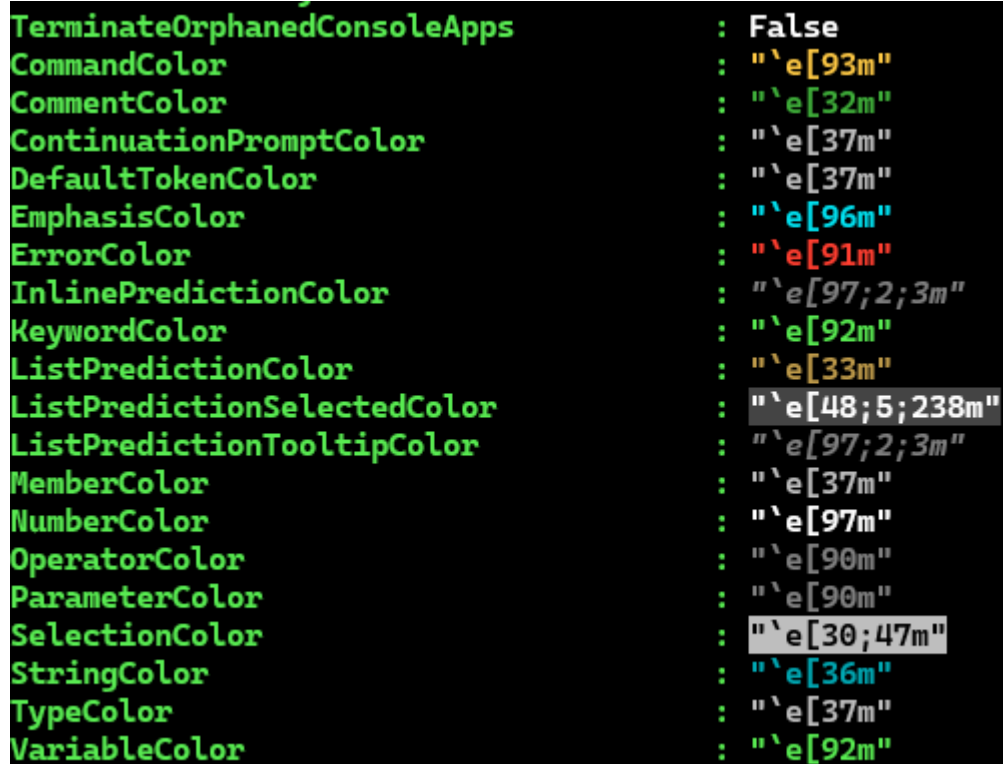
The export process will save your configurations settings to a JSON file in `$HOME`. When you import the module, if the file is found, the values will be imported and used in your PowerShell session. If you want to revert to the default settings, delete the JSON file, `$HOME\.pscalendarConfiguration.json`.

> If you uninstall the `PSCalendar` module, you will need to manually delete the JSON file.

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:

```
TerminateOrphanedConsoleApps       : False
CommandColor                       : "`e[93m"
CommentColor                       : "`e[32m"
ContinuationPromptColor            : "`e[37m"
DefaultTokenColor                  : "`e[37m"
EmphasisColor                      : "`e[96m"
ErrorColor                         : "`e[91m"
InlinePredictionColor              : "`e[97;2;3m"
KeywordColor                       : "`e[92m"
ListPredictionColor                : "`e[33m"
ListPredictionSelectedColor        : "`e[48;5;238m"
ListPredictionTooltipColor         : "`e[97;2;3m"
MemberColor                        : "`e[37m"
NumberColor                        : "`e[97m"
OperatorColor                      : "`e[90m"
ParameterColor                     : "`e[90m"
SelectionColor                     : "`e[30;47m"
StringColor                        : "`e[36m"
TypeColor                          : "`e[37m"
VariableColor                      : "`e[92m"
```

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 1/8/2025,15,8,2025

                  August 2025

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.

Integrating with PSReminderLite

You might be interested in integrating this module with the PSReminderLite PowerShell module. The `PSReminderLite` module is a lightweight reminder system that uses a SQLite database.

```
PS C:\> Get-PSReminder -Tag Event

ID   Event                    Comment                        Date    Countdown
--   -----                    -------                        ----    ---------
1164 PowerShell Summit                               4/7/2025 8:00 AM  8.16:53:32
1165 PowerShell Summit                               4/8/2025 8:00 AM  9.16:53:32
1166 PowerShell Summit                               4/9/2025 8:00 AM 10.16:53:32
1167 PowerShell Summit                              4/10/2025 8:00 AM 11.16:53:32
1173 Workplace Ninjas UK                             6/16/2025 9:00 AM 78.17:53:32
1174 Workplace Ninjas UK                             6/17/2025 9:00 AM 79.17:53:32
```

I have configured my PowerShell profile to use PSReminder dates as highlighted dates in the calendar. In my PowerShell `prompt` function, I have this code:

```
$days = (Get-PSReminder -All).Where({-Not $_.Expired}).Date.ToShortDateString() | Select-Object -Unique
$PSDefaultParameterValues.'*-*Calendar:HighlightDate' = $days
```

This ensures that I always have my upcoming reminders highlighted in the calendar without having to manually enter them.

```
PS C:\> $PSDefaultParameterValues.GetEnumerator() | where Name -match calendar

Name                       Value
----                       -----
*-*Calendar:HighlightDate  {4/1/2025, 4/2/2025, 4/7/2025, 4/8/2025…}
```

```
[29 Mar 15:16] PS7.5 C:\▶ Get-PSReminder -Month 6

ID    Event                       Comment                      Date    Countdown
──    ─────                       ───────                      ────    ─────────

1192  Cmdlet Working Group                              6/4/2025 12:00 PM 66.20:43:25
1173  Workplace Ninjas UK                               6/16/2025 9:00 AM 78.17:43:25
1174  Workplace Ninjas UK                               6/17/2025 9:00 AM 79.17:43:25
1193  Cmdlet Working Group                              6/18/2025 12:00 PM 80.20:43:25
1206  PSConfEU                                          6/23/2025 8:00 AM 85.16:43:25
1207  PSConfEU                                          6/24/2025 8:00 AM 86.16:43:25
1208  PSConfEU                                          6/25/2025 8:00 AM 87.16:43:25
1209  PSConfEU                                          6/26/2025 8:00 AM 88.16:43:25


[29 Mar 15:16] PS7.5 C:\▶ Get-Calendar -Month June

                    June 2025

 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     1     2     3     4     5
```

Related Modules

You might also be interested in the PSClock module which displays a WPF-base clock on your desktop as well as PSTimers which is a PowerShell module with a number of timer and countdown commands.

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.

Please post any issues or feature requests in the module's Issues.

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
```

Or, if you have the PSScriptTools module installed, you can use the `Get-ModuleCommand` function to get a list of all commands in a module.

Commands not supported on non-Windows platforms will be handled on a per-command basis.

Export-PSCalendarConfiguration

Synopsis

Save the current calendar configuration settings to a file.

Syntax

```
Export-PSCalendarConfiguration [-Passthru] [-WhatIf] [-Confirm] [<CommonParameters>]
```

Description

If you have customized the calendar configuration settings, and which to alway use them, use this command to export the settings to a JSON file in $HOME. The next time you import the module, the settings will be automatically imported. If you want to revert to the default settings, delete the JSON file, $HOME\PSCalendarConfiguration.json. If you uninstall the module, you will need to manually delete the JSON file.

This command was added in module version 2.10.0

Examples

Example 1

```
PS C:\> Export-PSCalendarConfiguration
```

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
```

-Passthru

Show the configuration settings file.

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

Required: False
Position: Named
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

System.Io.FileInfo

Notes

This command has an alias of Save-PSCalendarConfiguration.

Learn more about PowerShell: https://jdhitsolutions.com/yourls/newsletter

Related Links

Get-PSCalendarConfiguration

Set-PSCalendarConfiguration

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>]
```

quarter

```
Get-Calendar -Quarter <Int32> [[-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 -CalendarYear <Int32> [-HighLightDate <String[]>] [-FirstDay <DayOfWeek>] [-NoANSI] [-MonthOnly]  [<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
```

```
                    March 2025

Sun    Mon    Tue    Wed    Thu    Fri    Sat
 23     24     25     26     27     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      4      5
```

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

Example 2

```
PS C:\> Get-Calendar -start "3/1/2025" -end "5/1/2025"
```

Display monthly calendars from March to May, 2025.

Example 3

```
PS C:\> Get-Calendar December -HighLightDate 12/4/2025,12/25/2025,12/24/2025,12/31/2025

                   December 2025

Sun    Mon    Tue    Wed    Thu    Fri    Sat
 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      3
```

Display a month and highlight specific dates in color.

Example 4

```
PS C:\>  Get-Calendar August -FirstDay Monday -HighLightDate 8/12,8/18,8/22

                   August 2025

Mon    Tue    Wed    Thu    Fri    Sat    Sun
 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      5      6      7
```

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 7/1/2025 -end 9/1/2025  | 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 2025 -NoANSI -MonthOnly

              January 2025

  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 2025 -NoANSI | Out-File c:\work\2025.txt
```

Create a yearly calendar for 2025 and save the output to a text file.

Example 8

```
PS C:\> Get-Calendar -Quarter 2
```

Display the months for the second quarter of the current year. The months will be displayed in a single column.

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: 0
Default value: current month
Accept pipeline input: False
Accept wildcard characters: False
```

-Quarter

Specify a calendar year quarter to display.

```
Type: Int32
Parameter Sets: quarter
Aliases:

Required: True
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
```

-Year

Select a year for the specified month.

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

Required: False
Position: 1
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
```

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: https://jdhitsolutions.com/yourls/newsletter

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: https://jdhitsolutions.com/yourls/newsletter

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
     March 2025
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
```

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

Example 2

```
PS C:\> ncal April 2025
     April 2025
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
Fri      4 11 18 25
Sat      5 12 19 26
```

Get an ncal for April 2025. This example is using the ncal alias.

Example 3

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

    January 2025
Sun     5 12 19 26
Mon     6 13 20 27
Tue     7 14 21 28
Wed  1  8 15 22 29
Thu  2  9 16 23 30
Fri  3 10 17 24 31
Sat     4 11 18 25

    February 2025
Sun     2  9 16 23
Mon     3 10 17 24
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

    March 2025
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
...
```

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: <u>https://jdhitsolutions.com/yourls/newsletter</u>

Related Links

<u>Get-Calendar</u>

<u>Get-MonthName</u>

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](#)

[Export-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. This syntax 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.

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

Export-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>]
```

quarter

```
Show-Calendar [[-Year] <Int32>] [-HighLightDate <String[]>] [-FirstDay <DayOfWeek>] [-MonthOnly]
 -Quarter <Int32> [<CommonParameters>]
```

calyear

```
Show-Calendar -CalendarYear <Int32> [-HighLightDate <String[]>] [-FirstDay <DayOfWeek>] [-MonthOnly] [<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 2025 -HighLightDate 2/22/21
```

Display February 2025 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 not work in the PowerShell ISE.

Example 4

```
PS C:\> Show-Calendar -Month January -Year 2025 -MonthOnly

            January 2025

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 5

```
PS C:\> Show-Calendar -Quarter 2
```

Display the months for the second quarter of the current year. The months will be displayed in a single column.

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
```

-Quarter

Specify a calendar year quarter to display.

```
Type: Int32
Parameter Sets: quarter
Aliases:

Required: True
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
```

-Year

Select a year for the specified month.

```
Type: Int32
Parameter Sets: month, quarter
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
```

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: https://jdhitsolutions.com/yourls/newsletter

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, 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 can display as many different calendars as you want at the same time.

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 12/2024 -end 2/2025 -highlight 12/24/24,12/25/24,12/31/24,1/1/25,2/14/25 -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 = @{"7/4/2025"="4th of July";"7/14/2025"="Bastille Day";"7/22/2025"="Family Visit"}
PS C:\> Show-GuiCalendar -start 7/1/2025 -BackgroundImage c:\scripts\zazu.gif -HighLightDate $h
```

Display July 2025 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 3/1/2024. 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 12/25/2024,12/31/2024 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 1/1/2025.

```
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 has an alias of gcal.

Learn more about PowerShell: https://jdhitsolutions.com/yourls/newsletter

Related Links

Show-Calendar

Get-Calendar

Show-PSCalendarHelp

Synopsis

Display a help PDF file for the PSCalendar module.

Syntax

```
Show-PSCalendarHelp [-Online] [<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. Alternatively, you can use the -Online switch to open the README file in the modules Github repository using your default web browser.

Examples

Example 1

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

Parameters

-Online

Opens the PSCalendar README file in the Github repository.

```
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

None

Notes

Learn more about PowerShell: https://jdhitsolutions.com/yourls/newsletter

Related Links

https://github.com/jdhitsolutions/PSCalendar