Sumatra
Development
www.sumatra.com

The Sumatra Holiday cmdlet for Microsoft Exchange 2010

Last Version: October 2011
Product Version: 1.0

Sumatra
Development
www.sumatra.com

Putting Holidays into Microsoft Exchange 2010 Server-Side

Copyright

Table of Contents

Table of Figures

Calendars in Exchange

This guide explains how Exchange Administrators can insert Holidays or special events into the calendars of any -- or all -- users into Exchange 2010.

Why the Need for a Holiday cmdlet?

Microsoft Exchange does not provide an any easily automated way for an Exchange Administrator to add company holidays to all her users' calendars.

Sumatra's suHoliday Exchange Management Shell cmdlet allows Exchange administrators the option of inserting holidays server-side into user accounts.

What does suHoliday do that client-side .hol files in Outlook do not?

So glad you asked.

Although a .hol file contains all of the holidays, it requires administrators or end users add the .hol data manually, one calendar at a time. Most end user's won't do this; Exchange Admins don't have the time.

You want a tool that runs server-side and you can script. You want to configure your local holidays or special events from one location, and not have to distribute .hol files. You want to add holidays to all users' calendars at once when *you* are ready. You want to incorporate this tool into your new user mailbox provisioning process.

You want Sumatra's suHoliday tool!

suHoliday allows you to effectively block out dates when most users wouldn't be in the office, being proactive to your user base.

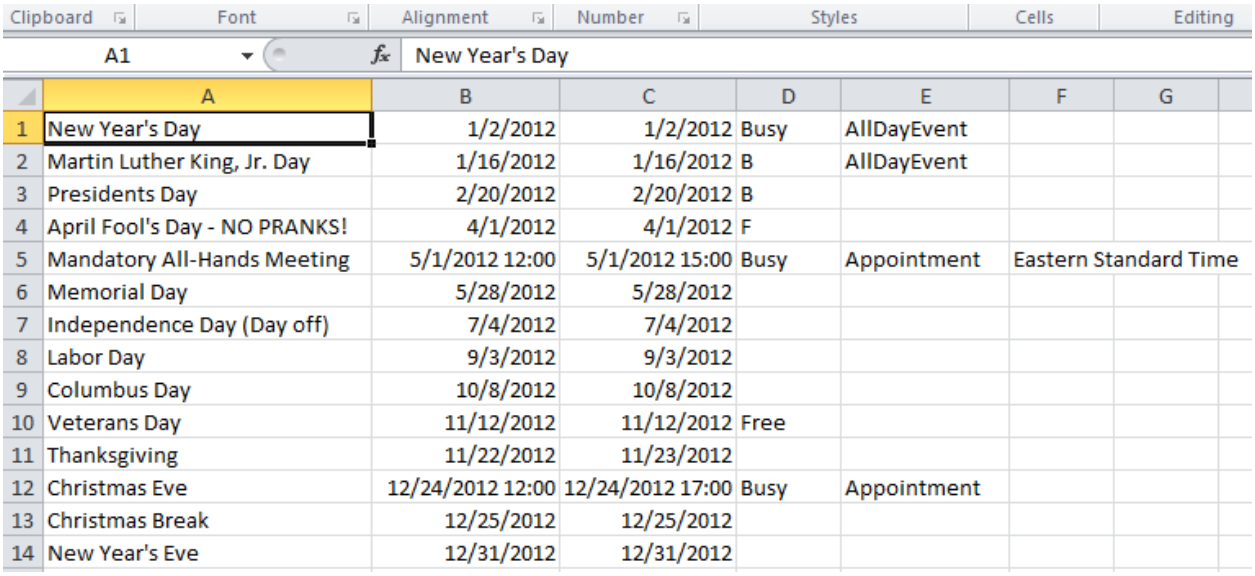
A glance at the following sample CSV file will give you a glimpse of suHoliday's capabilities.

But wait there's more:

- You can specify whether a holiday should be BUSY or FREE
- Holidays need not span a 24-hour period (so you could include an early-release day like Christmas Eve or block out your 3 hour shareholders meeting!
- Holidays can "float" with a time zone. For example, Independence Day starts and ends at midnight in all US time zones, or be "nailed" to a specific time zone. For example, your company-wide "all-

hands" meeting happens at ONE time in one time zone. All calendars need to reflect that fixed start-time, yet be intelligent enough to adjust the time for uses with calendars in different time zones.

Figure 1: Sample suHoliday input file



Clipboard | Font | Alignment | Number | Styles | Cells | Editing

A1 | fx | New Year's Day

	A	B	C	D	E	F	G
1	New Year's Day	1/2/2012	1/2/2012	Busy	AllDayEvent		
2	Martin Luther King, Jr. Day	1/16/2012	1/16/2012	B	AllDayEvent		
3	Presidents Day	2/20/2012	2/20/2012	B			
4	April Fool's Day - NO PRANKS!	4/1/2012	4/1/2012	F			
5	Mandatory All-Hands Meeting	5/1/2012 12:00	5/1/2012 15:00	Busy	Appointment	Eastern Standard Time	
6	Memorial Day	5/28/2012	5/28/2012				
7	Independence Day (Day off)	7/4/2012	7/4/2012				
8	Labor Day	9/3/2012	9/3/2012				
9	Columbus Day	10/8/2012	10/8/2012				
10	Veterans Day	11/12/2012	11/12/2012	Free			
11	Thanksgiving	11/22/2012	11/23/2012				
12	Christmas Eve	12/24/2012 12:00	12/24/2012 17:00	Busy	Appointment		
13	Christmas Break	12/25/2012	12/25/2012				
14	New Year's Eve	12/31/2012	12/31/2012				

Overview – Train Hard, Fight Easy!

Regardless of the number of users, servers, or additional engineering requirements you have, we recommend three things.

1. ALWAYS run your cmdlet on a test system BEFORE moving it into your production environment.

2. ALWAYS run your cmdlet on a test system BEFORE moving it into your production environment.

3. ALWAYS run your cmdlet on a test system BEFORE moving it into your production environment.

We cannot state enough the importance of testing prior to deployment. The most successful migrations we have seen have been the ones with the most preliminary testing. Our motto comes from Marshall Zhukov via the Navy SEALS: *Train hard, fight easy*.

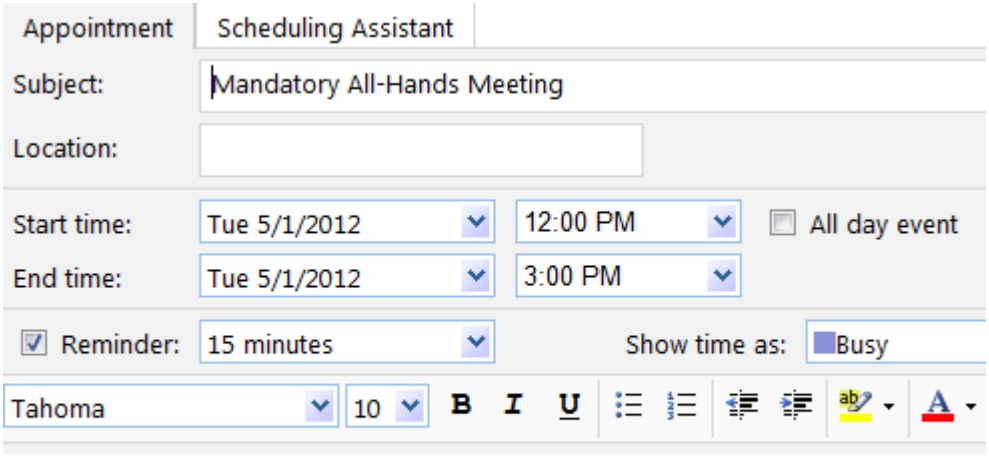
Limitations

To paraphrase Dirty Harry – you have to know your limitations.

Here are suHoliday's limitations:

- **It does not insert holidays past January 2, 2013**. Why? First – it's new and we do not want you to get in over our heads. Second, most organizations set their holiday/paid-time-off schedule a year at a time. Third, hey--- what on earth do you want for free/ ultra-low cost? If all goes well in the next few months we'll consider extending this – otherwise follow our blog for news about next year's version.
- **Inserted holidays contain a link to the Sumatra web site in the Agenda field.** Why? See the earlier comment on what do you want for free? If any well-heeled corporations out there want to private label this, please drop Sumatra a line.

Figure 2: Sample calendar item inserted by suHoliday



Added courtesy of the Exchange Calendaring experts: Sumatra Development

- This software is copyright Sumatra Development LLC, and is licensed software. It is subject to the terms in the license file. Please check them over. They are not onerous; we are not attempting to rip you off – but you certainly cannot claim that this code is yours, modify it, or redistribute it.
- suHoliday works on Exchange 2010. It does NOT work on prior versions of Exchange.

Configuration – Exchange 2010

Software and Admin Requirements

Make sure that your environment meets the following software requirements.

MS Exchange 2010 SP1

PowerShell 2.0

.NET Framework 4.x

The Microsoft Exchange Web Services Managed API v 1.1.

To execute the suHoliday cmdlet you will need a service account with Impersonate permissions. Permissions are the biggest issue we see with E2K10 installations.

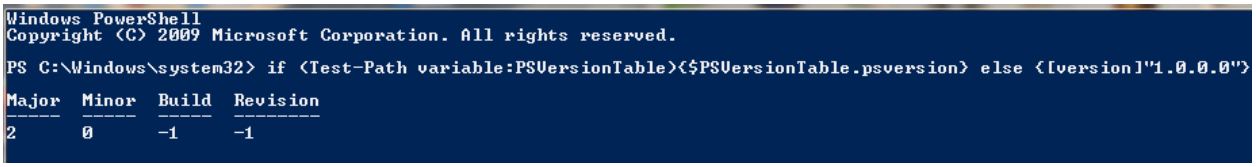
How to tell if you have PowerShell 2.0

Open PowerShell and run this on the command line:

```
if (Test-Path variable:PSVersionTable){$PSVersionTable.psversion} else {[version]"1.0.0.0"}
```

"Major" being "2" means you have Version 2, and are good to go.

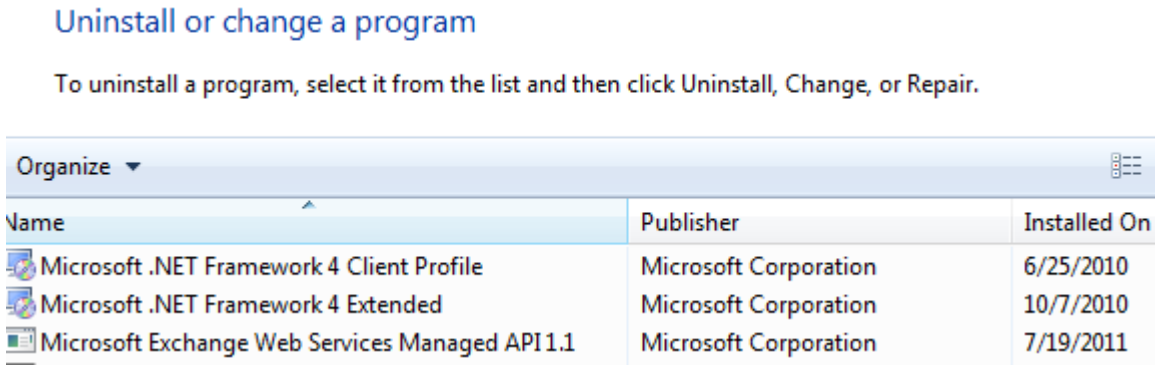
Figure 3: Determine if you have .Net Framework 4.0



How to tell if you have .NET Framework 4.0

Open your Control Panel and look at Programs and Features:

Figure 4: Determine if you have .Net Framework 4.0

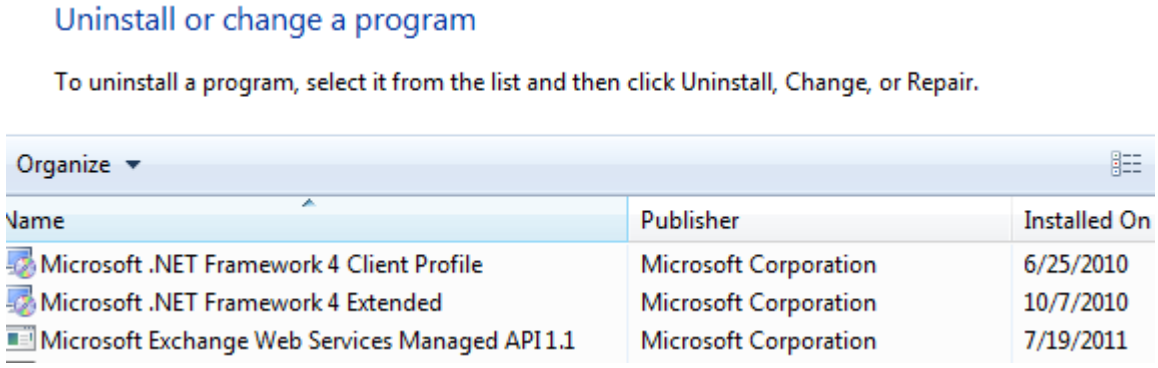


If not, refer to Microsoft's web site.

How to tell if you have EWS Managed API 1.1

Open your Control Panel and look at Programs and Features:

Figure 5: Determine if you have EWS Managed API 1.1



If not, download the API from Microsoft's web site.
http://www.microsoft.com/download/en/details.aspx?id=13480

suHoliday is scriptable

You will see PowerShell commands throughout this document used to setup, configure, and use the suHoliday cmdlet. You can cut and paste the lines shown in this document. You can also look at the script Add-Template to see how you can abstract all your crucial environment variables for adding holidays to user calendars.

Installing the cmdlet

We take you through this process explicitly instead of using an installer. Why? Much less mystery in it for you and there is a lot of rogue code out there – you deserve to see what is going where and control it yourself.

Create a folder "suHoliday" under your WindowsPowershell\Modules directory,

e.g.: C:\Users\administrator.COD\Documents\WindowsPowerShell\Modules\suHoliday

Copy the files from the zip file into that newly created Modules\suHoliday subdirectory. This makes the suHoliday cmdlet accessible to all users (on that machine.)

Advanced users: you can install the cmdlet in another directory, but you'llhave to supply an absolute path to the suHoliday.dll file when you want to use it.

Importing the cmdlet

Now you need to import the cmdlet – basically tell PowerShell that it's there and to take it in.

Open PowerShell and type[1]:

```
import-module suholiday.dll –verbose
```

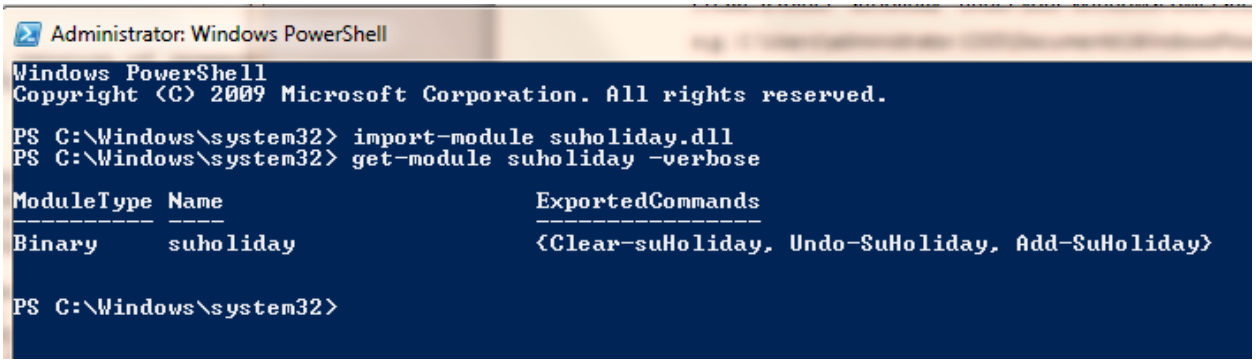
To verify this was successful, type:

```
get-module suholiday –verbose
```

If all goes well, this is what you will see:

Figure 6: Import-Module results

[1] **NOTE to advanced users**: IF YOU USE A DIRECTORY OTHER THAN "WindowsPowerShell\Modules\suHoliday" subfolder, you'll have to specify the absolute path when specifying the dll in the "import-module" cmdlet above.



To remove the module,

```
remove-module suholiday.dll
```

Set Credentials

Create a credential variable. **The credential userID MUST be the primary SMTP address of the service account that has impersonation rights.** The default credentials are not permitted.

```
$c=get-credential exsu@cod.sumatra.local
```

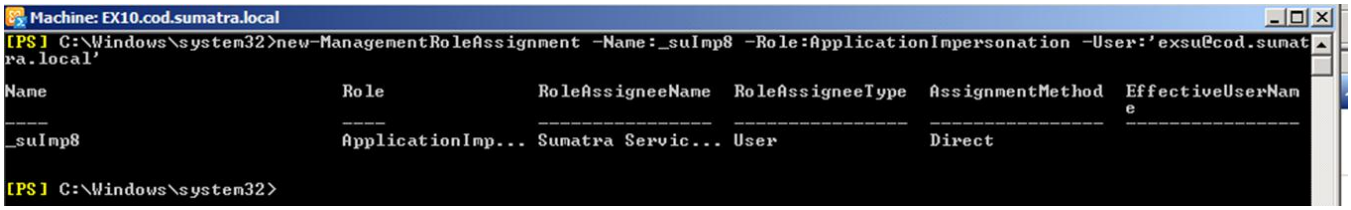
Remember, to add calendar items to all end user accounts using the service account, you have to grant that account impersonation rights, via ManagementRoleAssignment.

Exchange 2010: Set Impersonation Permissions

In Exchange 2010 you must create a new ManagementRoleAssignment for you service account ("_suImp8")

```
new-ManagementRoleAssignment
-Name:_suImp8
-Role:ApplicationImpersonation
-User:'exsu@cod.sumatra.local'
```

Figure 7: Setting Impersonation via ManagementRoleAssignment



Note on Permissions: suHoliday uses Impersonate

When do you use which permissions?

- Impersonate is typically used for ENABLED user accounts. Note, Impersonate fails when it tries to access a disabled account. **suHoliday uses Impersonate and works ONLY on User accounts.**
- Delegate is used when dealing with DISABLED accounts, such as ROOMS disabled end user mailbox accounts, or in environments with a Resource Forest Trust. Note: the actual mailbox permission is "FullAccess" (Full access is set via add-mailboxpermission command shell). **suHoliday does NOT use delegate access.**
- suHoliday will NOT work on DISABLED accounts. If you want another cmdlet to work on Resource accounts, please let us know!

Setting Impersonation Permissions for Office 365 / Live@Edu / Hosted Exchange

If you are in the Exchange Cloud, you need to know how to set up your permissions on your Office 365 or Live @ EDU Admin account so that the cmdlet has access to all accounts to do what it needs. You will use "Windows PowerShell Remoting." For those not that familiar with remoting, here are the steps:

1. Setup live@edu:
    a. Your domain must be Org-owned (you'll be setting RBACs)
    b. Create a service account, e.g. deleg8@livetest.YOURDOMAIN.com
2. On your PC, connect your local instance of Windows PowerShell to Outlook Live
    a. Prerequisites:
        i. You'll need Windows PowerShell 2.0.
            1. Get it here: http://support.microsoft.com/kb/968929
            2. If you have windows XP x86 must upgrade to SP3+;
            3. If you have windows XP x64 use the windows server 2003 x64 version
        ii. Launch PowerShell: Start > All Programs > Accessories > Windows PowerShell > Windows PowerShell right-hand click and "Run-As administrator"
    b. Set the credentials for your Windows Live ID and Password for your Outlook Live account, then define a session
        i. $LiveCred = Get-Credential
        ii. $Session = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri https://ps.outlook.com/powershell/ -Credential $LiveCred -Authentication Basic -AllowRedirection
            1. If it fails to execute, then:
                a. Set-ExecutionPolicy RemoteSigned
                b. …. Or unrestricted if you are brave

```
$LiveCred = Get-Credential
$Session = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri
https://ps.outlook.com/powershell/ -Credential $LiveCred -Authentication Basic -AllowRedirection
```

        iii. Start a session to import the Outlook Live commands into your session.
            1. Import-PSSession $Session
            2. Setup RBAC impersonation using built-in role "ApplicationImpersonation"

```
Import-PSSession $Session
```

3. Define RBAC impersonation in PowerShell (here “deleg8” is your Service Account for the migration). This is the same as it is for on-premises Exchange 2010:
    a. New-ManagementRoleAssignment -Name EWSImpersonation -Role ApplicationImpersonation -User deleg8@livetest.YOURDOMAIN.com

```
New-ManagementRoleAssignment -Name EWSImpersonation -Role ApplicationImpersonation -User
deleg8@livetest.YOURDOMAIN.com
```

    b. After you are done, disconnect Windows PowerShell from Outlook Live:
        i. Remove-PSSession $Session

```
Remove-PSSession $Session
```

Up-to-the-Minute Information

For more info not covered here please check out our blog at:

http://calendarservermigration.blogspot.com/

We put all information relevant to our calendar server migrations there as quickly as possible.

Using suHoliday

Add-suHoliday

The full cmdlet syntax is:

Figure 8: suHoliday Add-suHoliday syntax

Add-suHoliday -HolidayInputFile <String> -Credential <PSCredential> [-UserInputFile <String>] [-PrimarySMTPAddress <String>] [-ExchangeVersion <String>] [-CasURL <String>] [-TimeZone <String>] [-Verbose] [-Debug] [-ErrorAction <ActionPreference>] [-WarningAction <ActionPreference>] [-ErrorVariable <String>] [-WarningVariable <String>] [-OutVariable <String>] [-OutBuffer <Int32>] [-WhatIf] [-Confirm]

Let's break these down.

-HolidayInputFile	The name and location of the CSV file containing the holidays you want to insert. This field is required.
-Credential	Credentials for your service account. This is set as a variable and passed in. If not specified, you will be prompted for credentials. The userid must be the Primary SMTP address, in the form or userid@domain.com See earlier in the manual. This field is required.
-UserInputFile	A file that contains the primary SMTP address and the end user's default time zone. This is an optional field.
-PrimarySMTPAddress	Pass the primary SMTP address in directly. If omitted, this will come from either the piped "get-mailbox" OR from the –userInputfile
-ExchangeVersion	Choices are "2010", or "2010_SP1". The default is "2010"
-CasURL	The URL of your Exchange CAS server. Although suHoliday can use autodiscover look up each user's CAS Server, we recommend you do NOT use

	autodiscover. Why? Each autodiscover request can take 15 to 30 seconds. If you have many users, suHoliday will take forever to complete. We recommend you pick on CAS server and use the optional CAS parameter.
-TimeZone	The name of your time zone in the Microsoft naming standard. If you have one time zone, pass it in via the -timezone switch and use get-mailbox to feed SMTP addresses to suHoliday. Examples: "Eastern Standard Time", "Pacific Standard Time", "GMT Standard Time" See Appendix A for a full list. Default (in cases where it becomes relevant) is Eastern Standard Time (just because many of Sumatra's clients are on the Eastern coast of the USA). A list of time zones can be found in Appendix A **(do not create your own names!)**
-Verbose	Creates additional reporting output to aid in debugging. Try it on one user before you run it on the entire organization. In general, you should start small and work up to larger modifications just so you are confident and comfortable in what server-modifying tools do. You would be amazed at some of the war stories we have of what sites have tried with little or no preparation.
-Debug	Not used
-ErrorAction	If you see "File suholiday.dll cannot be loaded because the execution of scripts is disabled on this system. Please see "get-help about signing" for more details." For security reasons, PowerShell does not allow all scripts to run. To grant permission, first run PowerShell as Administrator, then use the cmdlet: Set-ExecutionPolicy RemoteSigned
-WarningAction	Not Used

Example for one user - when you know their time zone

As you would execute it:

Figure 9: Sample Add-suHoliday command for one user

```
Add-suHoliday -PrimarySMTPAddress russ@cod.sumatra.local -Credential $c  -ExchangeVersion 2010 -CAS
"https://ex10/ews/exchange.asmx" -HolidayInputFile
"C:\Users\administrator.COD\Desktop\suHoliday\2012Holidays.csv" -TimeZone "Eastern Standard Time"
```

Formatted so you can see the structure:

```
Add-suHoliday
-PrimarySMTPAddress russ@cod.sumatra.local
-Credential $c
-ExchangeVersion 2010
-CAS "https://ex10/ews/exchange.asmx"
-HolidayInputFile "C:\Users\administrator.COD\Desktop\suHoliday\2012Holidays.csv"
-TimeZone "Eastern Standard Time"
```

Example for all users when in the same time zone (piped user list from get-mailbox)

As you would execute it:

Figure 10: Sample Add-suHoliday command with user list piped from Get-Mailbox

```
get-mailbox -Filter {RecipientTypeDetails -eq "UserMailbox"} | select-object PrimarySMTPAddress | Add-suHoliday  -Credential $c  -ExchangeVersion 2010 -CAS "https://ex10/ews/exchange.asmx" -HolidayInputFile "C:\Users\administrator.COD\Desktop\suHoliday\2012Holidays.csv" -TimeZone "Eastern Standard Time"
```

Note: In this example, we use a filter to limit the mailbox list to those “UserMailboxes”

Formatted so you can see the structure:

```
get-mailbox
-Filter {RecipientTypeDetails -eq "UserMailbox"} | select-object PrimarySMTPAddress |
Add-suHoliday
-Credential $c
-ExchangeVersion 2010
-CAS "https://ex10/ews/exchange.asmx"
-HolidayInputFile "C:\Users\administrator.COD\Desktop\suHoliday\2012Holidays.csv"
-TimeZone "Eastern Standard Time"
```

For groups of users in different time zones

If you are an enterprise with users in multiple time zones the situation gets a little more intricate. If you assume they are all in the same time zone, the holiday “all-day-events” will span multiple days. You must in this case specify a time zone for each user.

We’ve built a PowerShell script that retrieves the default time zones for all of your users. Why? Because Microsoft’s cmdlets do not make it easy and fast.

1. Edit the script "getUserTimezones.ps1" and change the outputfile, the default time zone and the mailbox selection criteria. Also, note that the get-mailbox is filtering the results based upon RecipientTypeDetails. You can edit this as you see fit.

```
#Define your 'default' timezone (if none is set)
$myDefaultTimezone="Eastern Standard Time"

#Define the output file
$myOutputFile="userlist.txt"

#Define the list of User Accounts to process
$myMailboxList = get-mailbox -Filter {RecipientTypeDetails -eq "UserMailbox"}
| select-object Identity,PrimarySMTPAddress

#If file exists, delete the file
$fileExists=test-Path $myOutputFile
if ($fileExists -eq "True"){del $myOutputFile}
#Loop through list and get
foreach ($t in $myMailboxList) {
    $priSMTP=$t.PrimarySMTPAddress
    $xi=get-mailboxRegionalConfiguration -Identity $t.Identity
    if ($xi.TimeZone -eq $Null) {$tt=$myDefaultTimezone} Else
{$tt=$xi.TimeZone}
    write-output "$priSMTP $tt" >> $myOutputFile
}
write-output "Done!  see the file $myOutputFile"
```

2. Run the script "getUserTimezones.ps1"

   You can also edit the “userlist.txt” file and remove any accounts you don’t want!

3. Run suHoliday using the 'UserInputFile' switch (e.g., assume the file is called 'userlist.txt')

   C:\Users\administrator.COD\desktop\suHoliday\getUserTimezones.ps1

Figure 11: Sample Add-suHoliday command with user list from an input file

```
Add-suHoliday -UserInputFile "C:\Users\administrator.COD\Desktop\suHoliday\userlist.txt" -
Credential $c  -ExchangeVersion 2010 -CAS "https://ex10/ews/exchange.asmx" -HolidayInputFile
```

```
"C:\Users\administrator.COD\Desktop\suHoliday\2012Holidays.csv"
```

Undo-suHoliday

Hope for the best, plan for the worst. It's crucial have a "Plan B." UNDO is your "Plan B." Undo removes the suHoliday inserted items from your user's calendars. The full cmdlet syntax is:

Figure 12: Undo-suHoliday cmdlet Syntax

```
Undo-suHoliday -Credential <PSCredential> [-UserInputFile <String>] [-PrimarySMTPAddress <String>] [-ExchangeVersion <String>] [-CasURL <String>] [-Verbose] [-Debug] [-ErrorAction <ActionPreference>] [-WarningAction <ActionPreference>] [-ErrorVariable <String>] [-WarningVariable <String>] [-OutVariable <String>] [-OutBuffer <Int32>] [-WhatIf]  [-Confirm]
```

Undo will only remove the events inserted by suHoliday. But keep in mind it will remove ALL the events inserted by suHoliday. That is, if you have run suHoliday multiple times to insert holidays, UNDO will remove them all when invoked.

For one user example

As you would execute it:

Figure 13: Sample Undo-suHoliday for one user

```
Undo-suHoliday -PrimarySMTPAddress russ@cod.sumatra.local -Credential $c  -ExchangeVersion 2010 -CAS "https://ex10/ews/exchange.asmx"
```

Formatted so you can see the structure:

```
Undo-suHoliday
-PrimarySMTPAddress russ@cod.sumatra.local
-Credential $c
-ExchangeVersion 2010
-CAS "https://ex10/ews/exchange.asmx"
```

For multiple users (via get-mailbox) example

As you would execute it:

Figure 14: Sample Undo-suHoliday for users piped from Get-Mailbox

```
get-mailbox -Filter {RecipientTypeDetails -eq "UserMailbox"} | select-object PrimarySMTPAddress | Undo-suHoliday -Credential $c  -ExchangeVersion 2010 -CAS "https://ex10/ews/exchange.asmx"
```

Formatted so you can see the structure:

```
get-mailbox
-Filter {RecipientTypeDetails -eq "UserMailbox"} | select-object PrimarySMTPAddress |
Undo-suHoliday
-Credential $c
-ExchangeVersion 2010
-CAS "https://ex10/ews/exchange.asmx"
```

For multiple users (from a file) example

As you would execute it:

Figure 15: Sample Undo-suHoliday for users from an input file

```
Undo-suHoliday -UserInputFile "C:\Users\administrator.COD\Desktop\suHoliday\userlist.txt" -Credential $c  -ExchangeVersion 2010 -CAS "https://ex10/ews/exchange.asmx"
```

Formatted so you can see the structure:

```
Undo-suHoliday
-UserInputFile "C:\Users\administrator.COD\Desktop\suHoliday\userlist.txt"
-Credential $c
-ExchangeVersion 2010
-CAS "https://ex10/ews/exchange.asmx"
```

Clear-suHoliday

Once you are done adding holidays to your user calendars, you should definitely run the clear-suHoliday cmdlet. Clear-suHoliday removes the holiday structure the Sumatra tool creates (this design allows suHoliday to minimize the holiday file reads and speeds the process.) If you plan on using PowerShell for anything else this will give you the maximum resources for your further work.

The full cmdlet syntax is:

Figure 16: Clear-suHoliday cmdlet Syntax

```
Clear-suHoliday
```

PowerShell Script File Templates included with the suHoliday Distribution

We include a few pre-defined PowerShell script templates:

Add_Template_v1.1

This script is a template that aims to routinize adding holidays for a single user.

All environment-specific variables have been isolated and documented at the start of the template. You are encouraged to modify this to suit your needs or to write your own scripts. Please feel free to share your work with the wider community.

Get_User_TimeZones

This script is meant for a multi-time zone Exchange environment. Refer to section "For groups of users in different time zones" earlier in the manual. If you are attempting to add holidays for users across a range of time zones, running this first will generate a specific list of users and time zones you can then feed into Add-suHoliday.

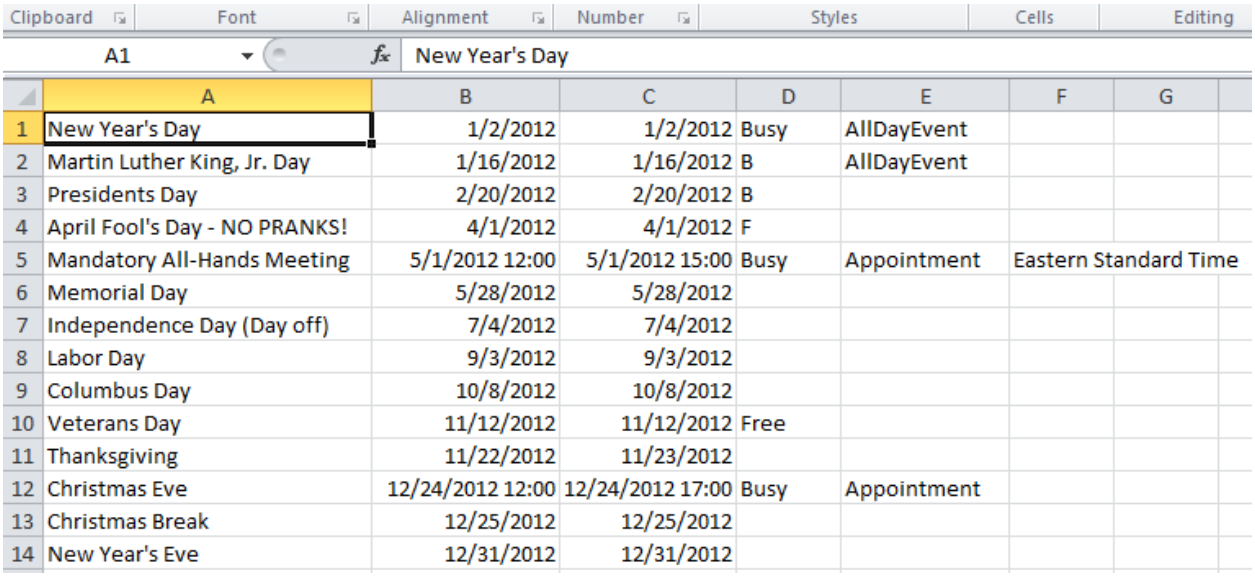
Why do this? Because it's MUCH faster than figuring out which user is in which time zone yourself, or forcing all holidays to be in one time zone (had have users in different time zones see holidays in their calendar span multiple days.)

Holiday File Structure

The Holiday file is a comma separated value file with six columns. The easiest way to edit it is either in Excel or a text editor. There are two sample CSVs included in the ZIP file, one for the US and another for the UK. If users wish to share others let us know and we'll see what we can do to smooth that process.

Here is a sample of the file in Excel, as well as a "text" file:

Figure 17: Sample Holiday File (shown in Microsoft Excel)

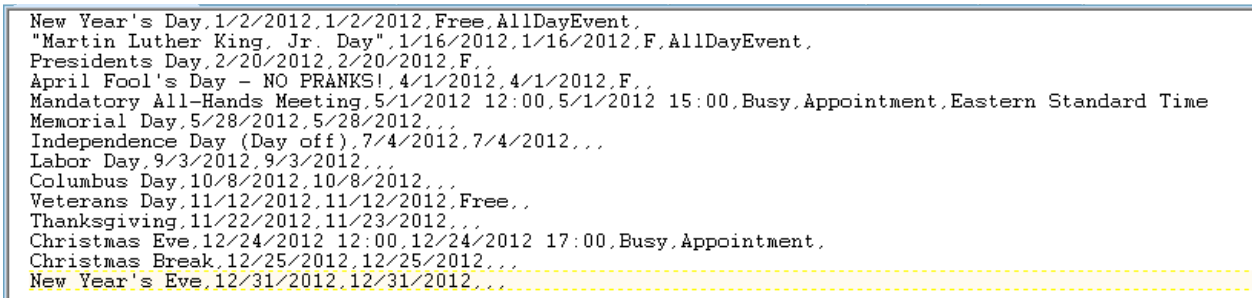


Clipboard | Font | Alignment | Number | Styles | Cells | Editing

A1 | New Year's Day

	A	B	C	D	E	F	G
1	New Year's Day	1/2/2012	1/2/2012	Busy	AllDayEvent		
2	Martin Luther King, Jr. Day	1/16/2012	1/16/2012	B	AllDayEvent		
3	Presidents Day	2/20/2012	2/20/2012	B			
4	April Fool's Day - NO PRANKS!	4/1/2012	4/1/2012	F			
5	Mandatory All-Hands Meeting	5/1/2012 12:00	5/1/2012 15:00	Busy	Appointment	Eastern Standard Time	
6	Memorial Day	5/28/2012	5/28/2012				
7	Independence Day (Day off)	7/4/2012	7/4/2012				
8	Labor Day	9/3/2012	9/3/2012				
9	Columbus Day	10/8/2012	10/8/2012				
10	Veterans Day	11/12/2012	11/12/2012	Free			
11	Thanksgiving	11/22/2012	11/23/2012				
12	Christmas Eve	12/24/2012 12:00	12/24/2012 17:00	Busy	Appointment		
13	Christmas Break	12/25/2012	12/25/2012				
14	New Year's Eve	12/31/2012	12/31/2012				

Figure 18: Sample Holiday File (shown in Microsoft Notepad)



```
New Year's Day,1/2/2012,1/2/2012,Free,AllDayEvent,
"Martin Luther King, Jr. Day",1/16/2012,1/16/2012,F,AllDayEvent,
Presidents Day,2/20/2012,2/20/2012,F,,
April Fool's Day - NO PRANKS!,4/1/2012,4/1/2012,F,,
Mandatory All-Hands Meeting,5/1/2012 12:00,5/1/2012 15:00,Busy,Appointment,Eastern Standard Time
Memorial Day,5/28/2012,5/28/2012,,,
Independence Day (Day off),7/4/2012,7/4/2012,,,
Labor Day,9/3/2012,9/3/2012,,,
Columbus Day,10/8/2012,10/8/2012,,,
Veterans Day,11/12/2012,11/12/2012,Free,,
Thanksgiving,11/22/2012,11/23/2012,,,
Christmas Eve,12/24/2012 12:00,12/24/2012 17:00,Busy,Appointment,
Christmas Break,12/25/2012,12/25/2012,,,
New Year's Eve,12/31/2012,12/31/2012,,,
```

Here are the details of the structure. Note that you can omit the column name from the file:

Figure 19: Holiday File Field Definitions

Col	Name	Description
A	Event Name	If the event name contains a comma in its name (e.g., MLK, Jr. Day), enclose it in double quotes. A spreadsheet usually does this automatically, a text editor does not.
B	Start Date / Time	If it is an all-day event, the time is assumed to start at midnight.

C	End Date / Time	If it is an all-day event, the time is assumed to end at midnight.
D	Free/Busy Status	Choices are Free, Busy, Tentative, and OOF. The default is Free. Either the entire word or the first letter will work (e.g., “Free”, “F”).
E	Event Type	Options are Appointment and All-Day Event. The default is All-Day-Event
F	Time Zone	This is meant for appointments which must be “nailed” to a specific time zone. For All-Day Events leave the time zone blank. It will assign a time zone to the event from either the “–Timezone” switch or from the Time zone passed in from the -UserInputFile switch. Also, use Microsoft names for the time zones as shown in Appendix A(i.e., do not create your own).

International Holiday Files

The suHoliday cmdlet supports international date formats as in this UK example:

Figure 20: Sample ‘International’ Holiday File

```
New Year's Day,1/1/2012,1/1/2012,Free,AllDayEvent,
Good Friday,6/4/2012,6/4/2012,F,,
Easter Monday,9/4/2012,9/4/2012,F,,
Mandatory All-Hands Meeting,1/5/2012 12:00,1/5/2012 15:00,Busy,Appointment,GMT Standard Time
May Day,7/5/2012,7/5/2012,,,
Spring Bank Holiday,4/6/2012,4/6/2012,,,
Spring Bank Holiday (Extra),5/6/2012,5/6/2012,,,
Summer Bank Holiday,27/8/2012,27/8/2012,,,
Christmas Day,25/12/2012,25/12/2012,,,
Boxing Day,26/12/2012,26/12/2012,,,
```

CAUTION: the date format of the system you are running the cmdlet from must agree with the date format of the data in your holiday file. This is another reason to start testing with one or two users on isolated systems before attempting this on your entire enterprise.

Appendix A: Microsoft Time Zone Names

Time Zone	*Time Zone*	*Time Zone*
Afghanistan Standard Time	Eastern Standard Time	Romance Standard Time
Alaskan Standard Time	Egypt Standard Time	Russian Standard Time
Arab Standard Time	Ekaterinburg Standard Time	SA Eastern Standard Time
Arabian Standard Time	Fiji Standard Time	SA Pacific Standard Time
Arabic Standard Time	FLE Standard Time	SA Western Standard Time
Atlantic Standard Time	Georgian Standard Time	Samoa Standard Time
AUS Central Standard Time	GMT Standard Time	SE Asia Standard Time
AUS Eastern Standard Time	Greenland Standard Time	Singapore Standard Time
Azerbaijan Standard Time	Greenwich Standard Time	South Africa Standard Time
Azores Standard Time	GTB Standard Time	Sri Lanka Standard Time
Canada Central Standard Time	Hawaiian Standard Time	Taipei Standard Time
Cape Verde Standard Time	India Standard Time	Tasmania Standard Time
Caucasus Standard Time	Iran Standard Time	Tokyo Standard Time
Cen. Australia Standard Time	Israel Standard Time	Tonga Standard Time
Central America Standard Time	Korea Standard Time	US Eastern Standard Time
Central Asia Standard Time	Mid-Atlantic Standard Time	US Mountain Standard Time
Central Brazilian Standard Time	Mountain Standard Time	Vladivostok Standard Time
Central Europe Standard Time	Mountain Standard Time (Mexico)	W. Australia Standard Time
Central European Standard Time	Myanmar Standard Time	W. Central Africa Standard Time
Central Pacific Standard Time	N. Central Asia Standard Time	W. Europe Standard Time
Central Standard Time	Namibia Standard Time	West Asia Standard Time
Central Standard Time (Mexico)	Nepal Standard Time	West Pacific Standard Time
China Standard Time	New Zealand Standard Time	Yakutsk Standard Time
Dateline Standard Time	Newfoundland Standard Time	
E. Africa Standard Time	North Asia East Standard Time	
E. Australia Standard Time	North Asia Standard Time	
E. Europe Standard Time	Pacific SA Standard Time	
E. South America Standard Time	Pacific Standard Time	

Sumatra Development

Like suHoliday? License it!

Go to http://www.sumatra.com/cmdlet.htm We ask a donation of US$20 per instance.

Don't like it? Tell us why. We're adults – we can take criticism.

What else does Sumatra offer?

Want to migrate legacy calendar data into Exchange server-side and preserve its links across users?

http://www.sumatra.com

Want to analyze conference room use in Exchange? Check out:

http://www.sumatraresourcewatch.com/resourcewatch4

Want to insert an ICS file into hosted Exchange? See:

http://www.sumatraresourcewatch.com/decaf

Look at our blog for all kinds of calendar-centric info:

http://calendarservermigration.blogspot.com/

Follow us on Twitter:

http://twitter.com/sumatra_dev

What else do you need?

We never learned anything listening to ourselves talk.

We only learn it when you folks tell us what you want.

If it involves calendaring technology, feel free to contact us!

We hear from customers regularly about calendaring-related issues. If you have a question or comment, feel frustrated at a calendaring-related problem, need our help, want to suggest new tools or invite us to lunch, please let us know!

Contact us!

The Managing Partners of Sumatra can be reached at:

zyg@sumatra.com

riuliano@sumatra.com

Check us out at www.sumatra.com as well as

Our blog:

http://calendarservermigration.blogspot.com/

And Twitter:

http://twitter.com/sumatra_dev