Dynamic Maintenance Windows

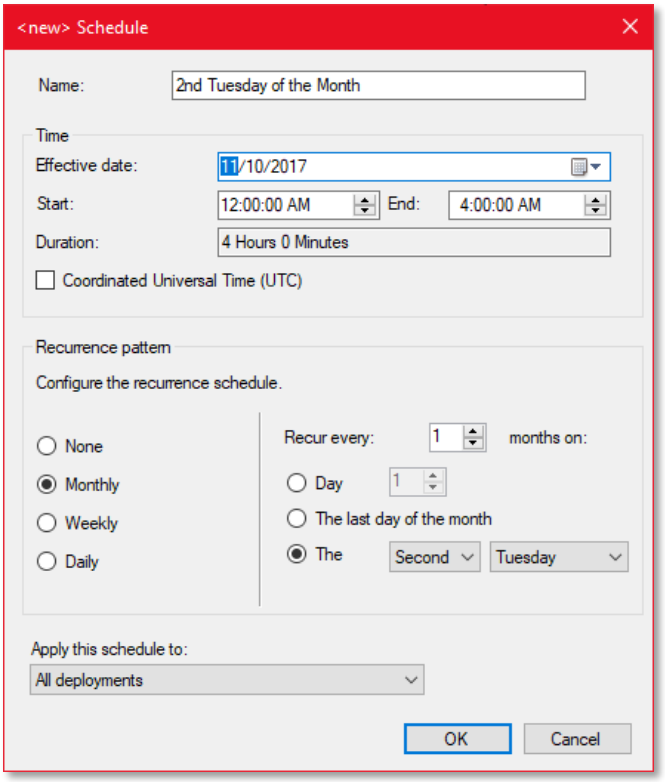
Overview

The second Tuesday of the month is Microsoft's "Patch Tuesday"; this is when they release the monthly hotfixes and Cumulative Updates. We build maintenance windows that Tuesday night, and the 3rd Thursday of the month, but what happens when the dates don't line up like they are expected to?

Issue

As explained, our patch cycle has two standard maintenance windows (MW).

Pilot Group: Installs the patches the night of Patch Tuesday, which is always the *second* Tuesday of the month; standard timeframe is 4 hours (0000-0400)
Production Group: Installs the patches the *third* Thursday of the month; standard timeframe is 4 hours (0000-0400)



The way that this is set up in ConfigMgr is a rather simplistic GUI. The GUI allows us to specify the Effective (Start) date, the start and end times, the monthly reoccurrence on day, "last day of the month" or on the "XY" day of the month. It also allows us to set what the schedule applies to: All deployments. Software Updates Only. Task Sequences Only.

In a perfect world, you can set it and forget it…but nothing is perfect. Next, let's take a look at an ideal example, and our roadblock.

Ideal:

Not ideal:

In the first screen shot, Patch Tuesday and our pilot Maintenance Windows (MW) are on the second week of the month. Our production MW is on the third week, as it should be. This gives us a week of testing the patches, so we know whether or not everything is "good".

If we look at the second screen shot, we can see that due to how the weeks are laid out in the month of November (2017), the 2nd Tuesday and 3rd Thursday are *in the same week*. That only gives us 1 day to actually test patching! Looking forwards, this happens a few times per *year*, so this isn't a random one off.

Our options are to turn off/cancel our current MW structure, or check every month, and create the MWs manually. What a pain…unless we can do it with PowerShell (Hint: Of course we can do it with PowerShell!).

Solution

The process is built off of 4 files. These files should be stored on the server, and *must live together in the same folder*:

- Window_Source.csv – Contains the details for the MWs you want to build, and on what collections
- Remove-CMMaintenanceWindow.ps1 – This removes any MWs on the collection(s) listed in the csv
- New-CMMaintenanceWindow.ps1 – This builds MWs based on what you specified in the csv
    - Source: https://gallery.technet.microsoft.com/scriptcenter/Setting-Maintenance-71f47c77
- MW-Feeder.ps1 – This reads the csv, and calls the above scripts in order.

Window_Source.csv:

The CSV must be in the following format. Explained below is what each column does:

CollectionID	Patch_Bucket	PlusDays	StartHour	StartMinute	HourDuration	MinuteDuration
COL0040C	MW_2Wed_0000_0400	1	0	0	4	0
COL00408	MW_3Thu_2300_0500	8	23	0	6	0

- CollectionID: This is the CollectionID for the collection you want to target
- Patch_Bucket: This is the name of the MW you are building. It can be whatever you want.
- PlusDays: How many days *after* Patch Tuesday, do you want the MW to be created for?
    - Based on Patch Tuesday
    - E.g.: +6 = Monday, +11 = Saturday. Negative values *do* work. -1 = Monday before Patch Tuesday
- StartHour: What hour do you want it to start?
    - Cannot exceed a value of 23
    - Must be in two digit, 24h, format
    - E.g.: "19", not "1900"
- StartMinute: What minute, within the hour do you want it to start?
    - Limited to 59 minutes. Anything over will add one hour and all else will be ignored.
    - E.g.: 61 = 1 hour, no minutes
- HourDuration: How long is the MW going to last, in hours?
    - 24h limitation imposed by SCCM
- MinuteDuration: How many minutes into the above hour?
    - Limited to 59 minutes. Anything over will add one hour and all else will be ignored.
    - E.g.: 61 = 1 hour, no minutes

Remove-CMMaintenanceWindow.ps1:

I added the code at the beginning, provided by MS, to load the SCCM module and connect to the Site. Remember to replace the site with your own sitecode, and the server with your server name.

New-CMMaintenanceWindow.ps1:

This script can remain unedited. It is already setup with everything it needs.

MW-Feeder.ps1:

This is the script that will kick off the others, in the appropriate order. Make sure to edit the sitecode and the server name.

How it all works:

1. Create/Edit your CSV file, containing the collections you want to edit, and with the maintenance window you want to build
2. Run the MW-Feeder.ps1 script; this does the following:
    a. Calls the Remove-CMMaintenanceWindow.ps1 script, which will remove *all* MWs on that collection
    b. Imports the data from the CSV
    c. Calls the New-CMMaintenanceWindow.ps1 script, which will dynamically build the new MW, based on Patch Tuesday, to your specifications

You can pick apart the New-CMMaintenanceWindow.ps1 if you want to tweak what it does, a bit more, but as it stands, this process works great and allows you to easily manage multiple MWs via (essentially) the CSV. Additionally, you can even set the script to run as a Scheduled task, monthly, so that you can automate the creation of the MWs.

CollectionID	Patch_Bucket	PlusDays	StartHour	StartMinute	HourDuration	MinuteDuration
COL0040C	MW_2Wed_0000_0400	1	0	0	4	0
COL00408	MW_3Thu_2300_0500	8	23	0	6	0

In the above example, the first Pilot collection (COL0040C) will have a MW created on the Wednesday (PlusDays: 1) directly after Patch Tuesday. It will start at 0000 in the morning (StartHour: 0), and will end at 0400 in the morning (HourDuration: 4).

The second, Prod, collection (COL00408) will start the Wednesday the *week after* Patch Tuesday (PlusDays: 8) at 2300, and go till 0500 in the morning on Thursday (HourDuration: 6).