

Repeating Events... Redux

Avoiding Scheduling Conflicts in the Long-Term

'Tis the season to pencil-in bookings and do some long-range planning. And if you should find conflicts, there is no better time to deal with them than now. If you haven't run into problems already with scheduling repeating events then you may not be aware of the considerable difficulty that can be involved, especially when using multiple methods for referencing dates. Let's start with a simple example.

S	M	T	W	T	F	S
				1	2	3
	5	6	7	8	9	10
	12	13	14	15	16	17
	19	20	21	22 Thanksgiving	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
		Saturday after Thanksgiving		Good		
		Sunday after Thanksgiving		Good		
		First Sunday in December		New, Good		
		Second Saturday in December		Good		
		Second Sunday in December		Good		

A new client would like to start a new annual tradition and wants to book the first Sunday in December, which is 12/3 in 2017. You take a look at your calendar and 12/3 is free (yellow, left). But, you're the cautious sort so you break out a 2018 calendar and mark all of this year's repeating events on it, too, just to be sure (right). Hey, alright, no conflicts here, either. But, some nagging feeling is bothering you, so you go crazy and pull up a 2019 calendar and fill it out (below right). And there it is, a conflict for the new gig, and

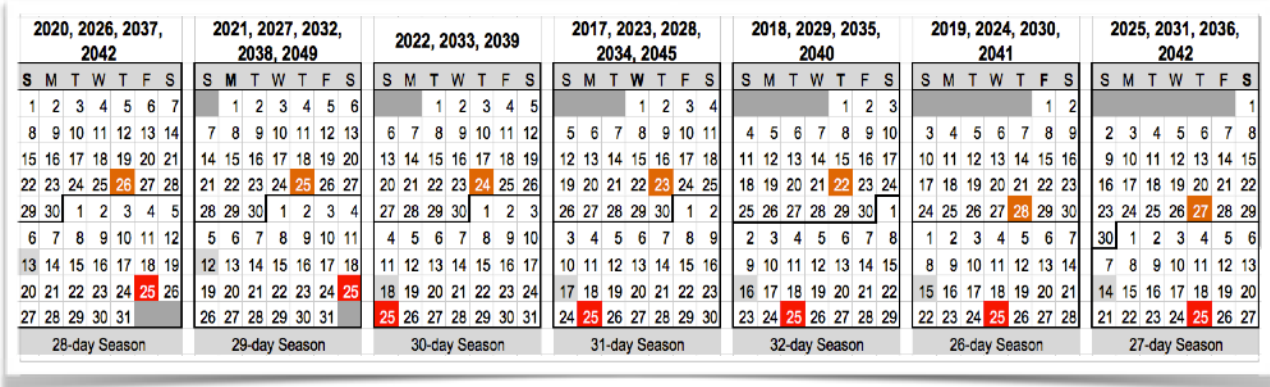
S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21 Thanksgiving	23	24	
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
		Saturday after Thanksgiving		Good		
		Sunday after Thanksgiving		Good		
		First Sunday in December		Good		
		Second Saturday in December		Good		
		Second Sunday in December		Good		

a possible problem for a gig you've been doing for years already! The Sunday after Thanksgiving is now the first Sunday in December, Yellow and Green events are on the same day! Also, when the town first hired you for the tree lighting and parade they said that they would need all your second Saturdays (tree lighting) and second Sundays (parade), but in 2019 the second Sunday is a week before the second Saturday. Fortunately, in this example, no one had booked the third Sunday and the parade can be safely moved and because of your foresight the town will know of the change two years in advance. And once the parade is moved your new booking can slide into the second Sunday slot.

You can see that this conflict could have easily gone unnoticed if we hadn't taken that extra step and filled out a 2019 calendar. And there could be more conflicts lurking. How many years worth of calendars do you think we'd have to fill out before covering all our bases? Would you believe, we'd have to do them through 2025! There must be an easier way! There is, let me show you.

S	M	T	W	T	F	S
					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 Thanksgiving	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
		Saturday after Thanksgiving				
		Sunday after Thanksgiving		Conflict		
		First Sunday in December		Conflict		
		Second Saturday in December		Reversed		
		Second Sunday in December		Reversed		
		Potential conflict if, say the tree lighting is supposed to be the night before the parade.				

Here is a graphic I use to visualize how dates rotate from year to year. There are 7 patterns for the dates around Christmas, each dictated by what day of the week November 1st falls on. In the leftmost calendar the first is on Sunday, the next is Monday and so on to represent every pattern possible for November and December. At the top I show some samples of years that use each pattern, but you'll notice that they are not uniform because of leap years... just one more thing that makes tracking repeating events difficult!



On each calendar (above), Thanksgiving is highlighted orange, and Christmas is red. One of the first things you'll notice is that "the season" (which I use to mean Black Friday to Christmas Eve) varies in length from 26- to 32-days long. It's this variability that can lead to trouble. The chart to the right shows the inconsistency of season lengths. For instance, there are no 27-day seasons until 2025, as alluded to in the example on page 1. You'd have to fill out calendars through 2025 to spot all possible conflicts using that system.

Year	1st On	Lgth
2017	Wed	31
2018	Thu	32
2019	Fri	26
2020	Sun	28
2021	Mon	29
2022	Tue	30
2023	Wed	31
2024	Fri	26
2025	Sat	27
2026	Sun	28
2027	Mon	29
2028	Wed	31
2029	Thu	32
2030	Fri	26
2031	Sat	27
2032	Mon	29
2033	Tue	30
2034	Wed	31
2035	Thu	32
2036	Sat	27
2037	Sun	28

The crux of our problem is just there, we make relative references to both fixed and unfixed dates: we base an event on Thanksgiving, or week of December, or Christmas, and it's all in flux every year. It is a rare thing indeed to have an event that makes an absolute reference to their date... almost no one wants to book December 11th every year since one year it will be on a Sunday, then a Tuesday, then a Thursday. Birthday parties are a possible exception, but even then folks are likely to have it on a weekend before or after the actual date. So, for the most part we are left with relative references for our dates: "we want the Saturday the week after Thanksgiving", or the second Sunday in December, or, my all time favorite, "can we do the Saturday before Christmas... unless it is later than the 20th then we want the second Saturday before Christmas"! And any one of these systems of reference *can* work and never give you a conflict, the problem is when different clients use different references. Even then, things may work for a while, as shown in the example on page 1.

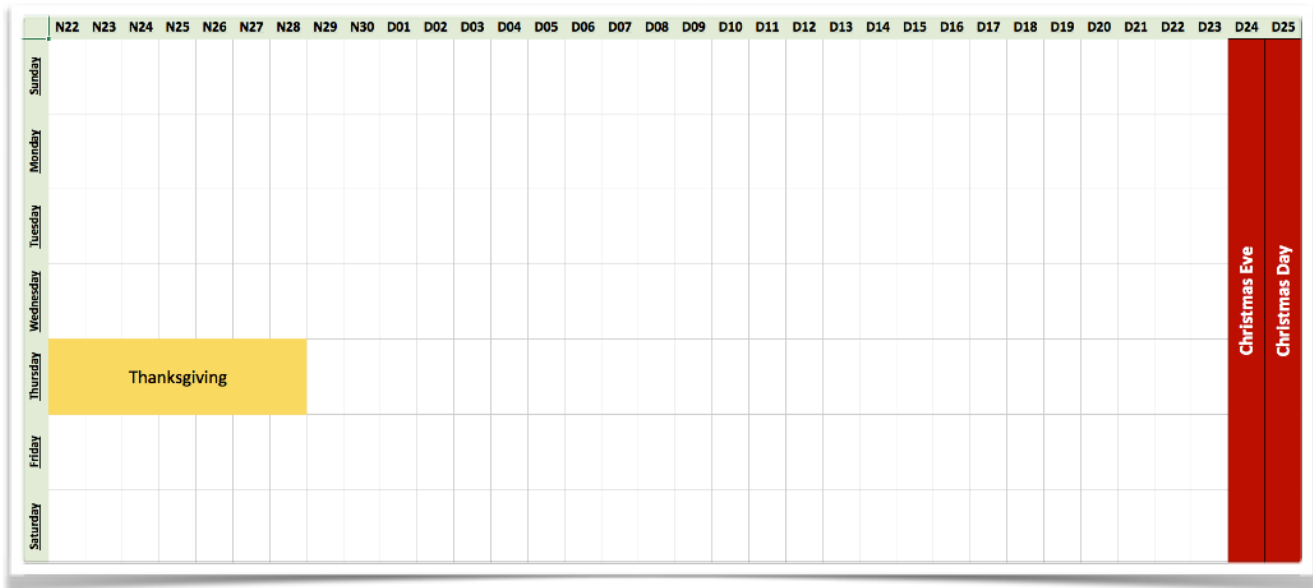
Eventually though, the 4th Sunday of the season, which is normally the 2nd Sunday before Christmas, will be the 1st Sunday before Christmas and you will have an unexpected conflict between clients who have happily coexisted on you calendar for three years!

Let's do an example. A new client wants the 3rd Sunday in December, so in 2017 that's the 17th. They want this date for years to come, so what more can we know about this date in other years? Well, to be the third of any weekday in a month it has to be at least the 15th (two weeks plus a day) and the latest it can be is the 21st (three full weeks). Now, if you have any other bookings that command Sun 12/15, Sun 12/16, Sun 12/17, Sun 12/18, Sun 12/19, Sun 12/20 or Sun 12/21 then at some point there will be a conflict.

So, how to remedy it? We need to figure out an absolute way to refer to the client's relative dates, and then an easy way to compare that range to the ranges of all your other repeating events... without having to fill out years and years worth of calendars! Let's use Thanksgiving as an example. The classic way to refer to when Thanksgiving occurs is the fourth Thursday in November, but if you want to know what date it falls on you're going to need to look

at that year's calendar. But, how else could we define it? If its the fourth Thursday that means at least three weeks have passed, so the earliest it can be in any year is the 22nd ($3 \times 7 + 1$). Further, it cannot fall in the fifth week of November, so the latest it can be is the 28th (4×7). So, Thanksgiving is the Thursday that falls in the range of November 22-28 on any given year. You can confirm this with the orange dates on the calendars above.

Below is a very useful, if odd looking, "calendar". On this calendar the dates are not tied to days of the week, instead the dates run left to right and the days of the week run top to bottom. What this allows us to do is see the whole season for **all** years at a glance. For instance, the red bars on the right are Christmas Eve and Christmas. They run vertically in columns D24 (December 24) and D25 (December 25) because they fall on that date every year, but that date can be Sunday (top row) or Monday (second row), and so on. Thanksgiving on the other hand, in gold, runs horizontally because it is always on a Thursday (fifth row). But not just any Thursday, it must fall no earlier than N22 (November 22) and no later than N28.

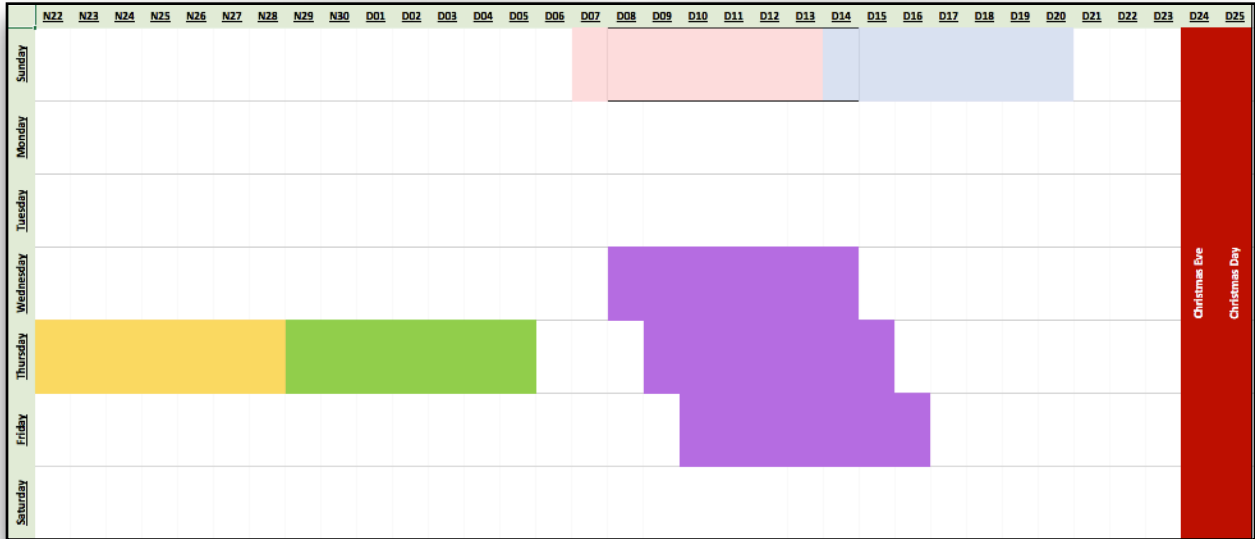


Using this system, even a convoluted request can be given an absolute reference. Try the one mentioned earlier: "the Sunday before Christmas unless it is later than the 20th then we want the second Sunday before Christmas(!)". This client would want the Sunday (first row) with a late date no later than 12/20 (D20). In order to cover all 7 possible calendar patterns (see page 2) we need to hold 7 consecutive days in the range, so the earliest early date would be the 6 days before 12/20. So we would reserve every Sunday that falls between 12/14–12/20 in any given year (shown in blue, below).

If you get another client who wants the 2nd Sunday of December you need to translate that into the new system: it becomes the only Sunday between 12/8–14 (second week means 7+1 to 7+7, or 8th to 14th). We go to our chart (below) to mark out 12/8–12/14 and oops, see that, there is a conflict! The new dates are shown with a box outline and that box overlaps an existing event. Your brunch from the previous example already commands every Sunday 12/14–20 (blue), and this new event, Sundays 12/8–14, overlaps on any Sunday the 14th. Now, that only happens once every several years, and as of this writing will not happen again until 2025 (you can check it on the calendar patterns on page 2), so you may have no issue booking it even knowing the conflict (I'll leave the ethical questions up to you). But, the main thing is that now you know where the hitch will be and you can plan for it with your clients. The simple solution in this case is to change the repeat of this new event to Sundays 12/7-13 (shown in pink, below).

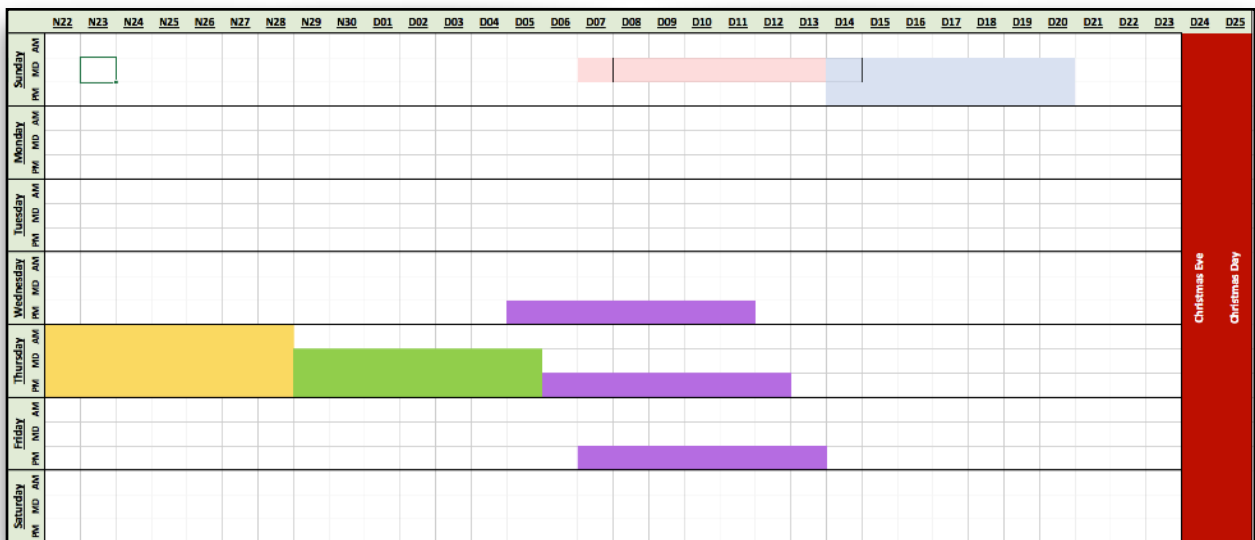
Let's try another: I have a client who always wants the Thursday after Thanksgiving. Let's translate that to the new system. Thanksgiving, as we saw above, is always 11/22–28, so the Thursday a week after that is 11/29–12/5.

To protect that client's booking you cannot book any Thursday that falls between 11/29 and 12/5. Example shown in green, below.



How about one final example: I have a client who books the second Wednesday of December and the following Thursday and Friday (which are not necessarily the second Thursday and Friday, but are sometimes the third). To be the second of any weekday in a given month the date must always fall between 12/8 (7 + 1) and 12/14 (7 + 7), so we would block out W 12/8–14, Th 12/9–15, F 12/10–16 (shown in purple). Now, this points out a problem. There are three Thursdays between the green event and the purple event that are not reserved, Thursdays 12/6–8. Go back to the chart on page 2 and you'll see that these dates are available this year and next, but not again until 2022! So, if you booked Thursday 12/7/2017, the client could rebook you for 2018, but basically never again (they'd have to get a fill-in for 4 years, guess how well that'd work out!). The better solution would be to ask your purple event if they would be willing to move their range 3 dates earlier (or 4 dates later). That would leave no gaps between Thanksgiving, the green event and the purple event, a solid booking!

Obviously, this graph is exceedingly simplistic. You can only book one event per day! A slightly more complex, but immensely more useful graph might divide each day into three broad time blocks: AM, Midday and PM. This would allow you to put three events per day into a graphical "repeats at a glance" chart, like this, showing the events from the previous calendar but placed in their appropriate timeslots:



You can see that this would be a much more useful tool, allowing you to see not only what days have openings and conflicts, but also get a better sense of what total percentage of your time is booked. Notice that I have shown the shift in the purple event as though the client approved the earlier date range. We still have a difficult to utilize gap for Sundays 12/21, 22 and 23 which will be hard to book unless they are just one-off type visits.

If you want to really get fancy, you can make a highly detailed “repeats at a glance” like I use. This is a closeup of mine:

You can see I have broken each day into hourly slots from 10am to 10pm (though 9–9 would probably be better). I have put years in the columns as a background so that I can tell when each weekday/date combination occurs, and repeated this background across the sheet so that when a new event covers part of it up I can just refer to an uncovered part. I have also outlined 7-date ranges based off of

	N12	N13	N14	N15	N16	N17	N18	N19	N20	N21	N22	N23	N24	N25	N26	N27	N28	N29	N30	D	
Sundays	10 AM	2017	2022	2021	2020	2025	2019								2018	2017	2022	2021	2020	2025	2024
	11 AM	2023	2033	2027	2026	2031	2024	Blessed Reflection													
	12 PM	2028	2039	2032	2037	2036	2030	Lafayette Village													
	1 PM	2034		2038		2042	2041														
	2 PM																				
	3 PM																				
	4 PM																				
	5 PM																				
	6 PM																				
	7 PM																				
Mondays	10 AM	2018	2017	2022	2021	2020	2025	2019	2018	2017	2022	2021	2020	2025	2019	2018	2017	2022	2021	2020	2025
	11 AM	2029	2023	2033	2027	2026	2031	2024							2024	2018	2017	2022	2021	2020	2025
	12 PM	2035	2028	2039	2032	2037	2036	2030													
	1 PM	2040	2034		2038		2042	2041													
	2 PM																				
	3 PM																				
	4 PM																				
	5 PM																				
	6 PM																				
	7 PM																				
Tuesdays	10 AM	2019	2018	2017	2022	2021	2020	2025	2019	2018	2017	2022	2021	2020	2025	2019	2018	2017	2022	2021	2020
	11 AM	2024	2029	2023	2033	2027	2026	2031	2024							2024	2018	2017	2022	2021	2020
	12 PM	2030	2035	2028	2039	2032	2037	2036	2030												

Thanksgiving so that I am reminded that it is better to keep to those ranges so as to avoid unbookable gaps. Below is my full “repeats at a glance” calendar. It is sized to fill my monitor so that all needed info is there at one shot. I have found this system to work very well for me in the 5 years or so that I’ve been using it, I hope you will find this info helpful to divide a system that works for you as well.