LEAP WEEKS DISTRIBUTION DURING 400-YEAR SPAN

This is in continuation of my: http://www.brijvij.com/brij8019_ln-yr.pdf
Following my divide 6 plan, for 400-years, first naturally div.6 year is 2004; and 2007th is the FIRST Additional Leap Week Year [(2000-80)=1920+87th =2007], the year as at: http://www.brijvij.com/bbv_Gen8Cal.doc. Following Years are planned to carry 53rd Week as Leap Week of the Year, between Y2000 thro 2399:

2000	2001	2002	2003	**2004**	2005	2006	**2007**	2008	2009	
2010	2011	2012	2013	2014	2015	**2016**	2017	2018	2019	
2020	2021	**2022**	2023	2024	2025	2026	2027	**2028**	2029	
2030	2031	2032	2033	**2034**	2035	2036	2037	2038	2039	
2040	2041	2042	2043	2044	2045	**2046**	2047	2048	2049	
2050	2051	**2052**	2053	2054	2055	2056	2057	**2058**	2059	
2060	2061	2062	2063	**2064**	2065	2066	2067	2068	2069	
2070	2071	2072	2073	2074	2075	**2076**	2077	2078	2079	
2080	2081	**2082**	2083	2084	2085	2086	**2087**	**2088**	2089	
2090	2091	2092	2093	**2094**	2095	2096	2097	2098	2099	+16+2Addl.
2100	2101	2102	2103	2104	2105	**2106**	2107	2108	2109	
2110	2111	**2112**	2113	2114	2115	2116	2117	**2118**	2119	
2120	2121	2122	2123	**2124**	2125	2126	2127	2128	2129	
2130	2131	2132	2133	2134	2135	**2136**	2137	2138	2139	
2140	2141	**2142**	2143	2144	2145	2146	2147	**2148**	2149	
2150	2151	2152	2153	**2154**	2155	2156	2157	2158	2159	
2160	2161	2162	2163	2164	2165	**2166**	**2167**	2168	2169	
2170	2171	**2172**	2173	2174	2175	2176	2177	**2178**	2179	
2180	2181	2182	2183	**2184**	2185	2186	2187	2188	2189	
2190	2191	2192	2193	2194	2195	**2196**	2197	2198	2199	+17+1Addl.
2200	2201	**2202**	2203	2204	2205	2206	2207	**2208**	2209	
2210	2211	2212	2213	**2214**	2215	2216	2217	2218	2219	
2220	2221	2222	2223	2224	2225	**2226**	2227	2228	2229	
2230	2231	**2232**	2233	2234	2235	2236	2237	**2238**	2239	
2240	2241	2242	2243	**2244**	2245	2246	**2247**	2248	2249	
2250	2251	2252	2253	2254	2255	**2256**	2257	2258	2259	
2260	2261	**2262**	2263	2264	2265	2266	2267	**2268**	2269	
2270	2271	2272	2273	**2274**	2275	2276	2277	2278	2279	
2280	2281	2282	2283	2284	2285	**2286**	2287	2288	2289	
2290	2291	**2292**	2293	2294	2295	2296	2297	**2298**	2299	+17+1Addl
2300	2301	2302	2303	**2304**	2305	2306	2307	2308	2309	
2310	2311	2312	2313	2314	2315	**2316**	2317	2318	2319	
2320	2321	**2322**	2323	2324	2325	2326	**2327**	**2328**	2329	
2330	2331	2332	2333	**2334**	2335	2336	2337	2338	2339	
2340	2341	2342	2343	2344	2345	**2346**	2347	2348	2349	
2350	2351	**2352**	2353	2354	2355	2356	2357	**2358**	2359	
2360	2361	2362	2363	**2364**	2365	2366	2367	2368	2369	
2370	2371	2372	2373	2374	2375	**2376**	2377	2378	2379	
2380	2381	**2382**	2383	2384	2385	2386	2387	**2388**	2389	
2390	2391	2392	2393	**2394**	2395	2396	2397	2398	2399	+16+1Addl

[Thus, there are **66 LWks** + (2007th , 2087th ,2167th , 2247th, & 2327th] **5 Addl.LWks** i.e. a total of 71 Leap Weeks between years Y2000 and Y2399; in a span of 400-years.

Reference: http://www.hermetic.ch/cal_stud/hlpwk/hexades.htm

The Hexade Pattern of the Hermetic Leap Week Calendar

This table shows the distribution of short (5-year) and long (6-year) hexades for the years 0 through 399. The year number is the number of the year which begins the hexade. The same pattern is repeated every 400 years, so this is also the pattern for the years 2000 through 2399.

Hexade number	Year number	Hexade type		Hexade number	Year number	Hexade type		Hexade number	Year number	Hexade type
1	0	short		2	5	long		3	11	long
4	17	short		5	22	long		6	28	long
7	34	short		8	39	long		9	45	long
10	51	short		11	56	long				
12	62	short		13	67	long		14	73	long
15	79	short		16	84	long		17	90	long
18	96	short		19	101	long		20	107	long
21	113	short		22	118	long				
23	124	short		24	129	long		25	135	long
26	141	short		27	146	long		28	152	long
29	158	short		30	163	long		31	169	long
32	175	short		33	180	long				
34	186	short		35	191	long		36	197	long
37	203	short		38	208	long		39	214	long
40	220	short		41	225	long				
42	231	short		43	236	long		44	242	long
45	248	short		46	253	long		47	259	long
48	265	short		49	270	long		50	276	long
51	282	short		52	287	long				
53	293	short		54	298	long		55	304	long
56	310	short		57	315	long		58	321	long
59	327	short		60	332	long		61	338	long
62	344	short		63	349	long				
64	355	short		65	360	long		66	366	long

67	372	short		68	377	long		69	383	long
70	389	short		71	394	long				

The 26 short hexades are as evenly spaced in the sequence of 71 hexades as is possible consistent with the properties of the calendar.

The pattern may be expressed as follows:

```
3x(sh,l,l,sh,l,l,sh,l,l,sh,l) + (sh,l,l,sh,l,l,sh,l) +
2x(sh,l,l,sh,l,l,sh,l,l,sh,l) + (sh,l,l,sh,l,l,sh,l)
```

Reference: http://www.go2zero.com/rwc/rwc.html

Bob McClenon's Reformed Weekly Calendar

In 1996, Bob McClenon introduced an idea for a version of a Weekly Calendar. On June 14, 1999, he elaborated on this a bit in a post he made to two e-mail discussion groups (Talk 2000 and the CALENDR-L mailing list). The full text of his idea can be found at the Talk 2000 archives.

Here's an excerpt from his presentation where he outlines the way the calendar is constructed:

> ***Both of the criticisms of the Comte thirteen-month calendar can be resolved by combining the concept of unequal but periodic months from the World Calendar with the concept of a leap week from the Colligan calendar. I have not read a proposal for such a calendar. I proposed such a calendar in 1996, prior to discovering the Colligan calendar. Jay Gary has labeled my proposal as the McClenon calendar, but I would prefer to give it a less personal name such as the Reformed Weekly Calendar. In the Reformed Weekly Calendar there will be twelve months, which are grouped into quarters, each consisting of a 31-day month and two 30-day months. The year consists either of 364 days (the twelve months) or 371 days (the twelve months and a leap week). The features of the Reformed Weekly Calendar are:***
>
> ***5a. The year begins on the first day of the first month, which may be 1 January if traditional month names are retained.***
>
> ***5b. The year is divided into twelve months of unequal length but following a regular pattern, because they are organized into quarters of a 31-day month followed by two 30-day months, with or without aleap week.***
>
> ***5c. The days are designated by weekdays, Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday, which repeat in a seven-day cycle. The cycle of the weekdays is directly related to the cycle of the days in the months because the combination of the length of the months and the leap week maintain that cycle.***
>
> ***5d. The year consists of either 364 or 371 days. In 329 years out of 400, it consists of 364 days. In 71 years out of 400, it consists of 371 days. In so-called leap or long years, a lea week is added.***
>
> ***5e. In years in which the year number is not divisible by 5, the year has 364 days. In years in which the year number is divisible by 5 but not by 50, the year has 371 days. In***

years in which the year number is divisible by 50 but not by 400, the year has 364 days. In years in which the year number is divisible by 400, the year has 371 days.

The Reformed Weekly Calendar has several advantages. Like the Comte thirteen-month calendar, the World Calendar, and the Colligan calendar, it ensures that the same day of the month always occurs on the same day of the week. Like the Colligan calendar, but unlike the Comte thirteen-month calendar or the World Calendar, the unbroken succession of the week is maintained so that one day out of seven can be set aside as a day of collective worship. Like the World Calendar, it makes only minor adjustments to the months, so that most anniversary dates can be maintained, and so that only a few new dates are introduced.

I wrote a computer program to generate the calendar based on the rules provided. Here, I printed it out for 401 years beginning with the Year 2000 (this file is almost 2 megabytes). For the begining of each month, I give the Gregorian equivalent and the Julian Day (the Julian Day actually begins at noon of Gregorian day).

A fundamental issue ***not*** addressed by McClenon (in his initial outline) is inauguration. Never mind, for the moment, how this idea could become popular enough to take hold, but when would it start?

There are some important issues involved:

1. Should year numbering be continued unbroken?
2. Should days-of-the-week pattern remain unbroken?
3. Should month-to-month pattern be continued unbroken?
4. Should the calendar begin with the first day of the week?

For now, I assume the answers to 1 - 3 above are positive. The 4th question might be yes. It is somewhat more critical than for the Gregorian (is didn't start on the first day of the week). The reason it makes a bigger difference for this calendar is that once it starts, the day of the week is fixed for every day of the month.

For example, if it were started Jan 1, 2000 (as in the print-out given above), Christmas would be on a Sunday--not just for this year, but forever! New Year's day would always happen on Saturday. This might have the effect of making Saturday the [de facto] first day of the week. July 4th would always be on Tuesday, and so on.

If you accept a "yes" answer to 1 - 4 above, then the calendar can only be inaugurated for a year that begins with Sunday. The next one would be 2006. After that, there's 2012, 2017, 2023, 2034, and so on. But in all these cases, all the days would be fixed in the same way. Remember the old nursery rhyme? It might take on a new meaning. Fine I suppose if you're born on one of the good days, but what about "Wednesday's child is full of woe, Thursday's child has far to go." As it is, everybody remembers their birthday as a day of the month. How many remember the day of the week? With the Reformed Weekly Calendar, children born then would never forget because they'd be reminded of it every year.

A main goal of our current civil calendar (a.k.a. Gregorian) is to keep each day in the season hapening at the same point on the calendar year-after-year. It is quite good in this regard. For example, the vernal equinox happens on the 21st of March every year (plus or minus a day or so). The Reformed Weekly Calendar wanders a bit. After the 400 year cycle, it ends up at the same point as the Gregorian, but in the mean time, the seasons wander.

Instead of "plus or minus a day or so" it's more like "plus or minus about 9 days." Here's a spread sheet I typed-up showing how it moves around and how it depends on the year of inauguration. In effect, the first day of spring would be moved. This might also impact the moveable holidays such as Easter.

Above, I gave a link to the calendar beginning with 01/01/2000. Here are some more possibilities. All of these are about 75k zipped with pkzip 2.04g.

- Sunday begining
  - 2006
  - 2012
  - 2017
  - 2023
  - 2034
  - 2045 (compare this one with the previous ones)
- Monday beginning
  - 2001
  - 2007
- The ones with the seasons shifted the most
  - 2036
  - 2285
  - 2325
  - 2365
  - 2370

If for some strange reason you'd like to see the calendar from some other starting point, let me know and I'll send it to you. Or you can generate your own using this computer code I wrote (in dBASE).

```
** these are quick and dirty routines I did just for the heck of it.  Use at your
own risk!
** this first one is a controlling program that I used to create and zip up all the
versions.


SET TALK OFF
SET CONSOLE OFF
SET SAFETY OFF
CLOSE DATABASES

nx = 1
DO WHILE nx <=500
   CLOSE ALL
   USE BOB EXCLU
   myr = yr + 1
   DELETE
   PACK
   USE gdates
   g1 = "12/31/" + LTRIM(STR(myr - 1))
   LOCATE FOR gregdate = ctod(g1)
   newjd = LTRIM(STR(jd))

   USE TEMP8 EXCLU
```

```
   ZAP
   APPEND FROM bobcal.prg SDF

   GO 26
   sdate = "01/01/" + LTRIM(STR(myr))
   newdaytrack = DOW(ctod(sdate))
   REPLACE f1 with LEFT(f1,18)+ STR(newdaytrack)

   GO 30
   REPLACE f1 with LEFT(f1,18) + newjd

   GO 63

   REPLACE f1 WITH LEFT(f1,27) + STR(myr)

   go top
   set alternate to temp8.prg
   set alternate on
   SCAN
      ? f1
   ENDSCAN
   set alternate off
   set alternate to
   close databases
   compile temp8.prg
   do temp8
   DO simp
   cmyr = LTRIM(STR(myr))
   !pkzip &cmyr temp1.txt
   nx = nx + 1
ENDDO



************************************************************
** I called this one simp.prg (for some reason I forgot).
** It crunches the stats
** Here's the structure for the table it uses
* Structure for database: TEMP3.DBF
* Field  Field Name  Type       Width    Dec    Index
*     1  F1          Character     50              N
*     2  D1          Date           8              N
*     3  JD          Numeric        8              N
*     4  NDAY        Character      2              Y
* ** Total **                      69
*
**** This one holds the final tally
* Structure for database: MONBEGIN.DBF
* Field  Field Name  Type       Width    Dec    Index
*     1  MON         Character      3              N
*     2  INAUG       Character      4              N
*     3  DY          Character      3              N
*     4  F16         Numeric        3              N
*     5  F17         Numeric        3              N
*     6  F18         Numeric        3              N
*     7  F19         Numeric        3              N
*     .... and so on until
*    19  F31         Numeric        3              N
*    20  F01         Numeric        3              N
*    21  F02         Numeric        3              N
*   ... and so on until F15
************************************************************
```

```
SET SAFETY OFF
SET CENTURY ON
SET TALK OFF
CLOSE DATABASES
mvar = "   "
DECLARE mons[13]
mons[1]  = "January"
mons[2]  = "February"
mons[3]  = "March"
mons[4]  = "April"
mons[5]  = "May"
mons[6]  = "June"
mons[7]  = "July"
mons[8]  = "August"
mons[9]  = "September"
mons[10] = "October"
mons[11] = "November"
mons[12] = "December"
mons[13] = "Leap"

DELETE FILE TEMP2.DBF
DELETE FILE TEMP2.MDX
USE temp3 EXCLUSIVE
COPY TO TEMP2
SELECT 1
USE TEMP2 EXCLUSIVE
SELECT 2
USE monbegin EXCLUSIVE
SELECT 1
APPEND FROM temp1.txt SDF
GO TOP
DELETE FOR .NOT. LEFT(LTRIM(f1),4) $ "Janu Febr Marc Apri May June July Augu Sept
Octo Nove Dece Leap " .AND..NOT. "Gregorian" $ f1
PACK

GO TOP
DO WHILE .NOT. EOF()
   mvar = RIGHT(RTRIM(f1),4)
   mvar2 = " " + SUBSTR(LTRIM(f1),AT(" ",f1), RAT(" ",RTRIM(f1)))
   SKIP
   REPLACE f1 WITH SUBSTR(f1,18,6) + mvar + SUBSTR(f1,26,13) + mvar2
   REPLACE d1 WITH CTOD(LEFT(f1,10))
   REPLACE jd WITH VAL(SUBSTR(f1,15,1)+SUBSTR(f1,17,3)+SUBSTR(f1,21,3))
   REPLACE nday WITH SUBSTR(f1,4,2)
   SKIP
ENDDO
GO TOP
DELETE FOR MOD(RECNO(),2)=1
PACK
i = 1
DO WHILE i <= 13
   mtag = "O" + LEFT(UPPER(mons[i]),3)
   mmons = mons[i]
   INDEX ON nday for "&mmons" $ f1  TAG &mtag
   i = i + 1
ENDDO

INDEX ON nday TAG nday
SET ORDER TO
GO TOP

nyear = YEAR(d1)
```

```
i = 1
DO WHILE i <=13
   mtag = "O" + LEFT(UPPER(mons[i]),3)
   SET ORDER TO &mtag
   GO TOP
   SELECT 2
   APPEND BLANK
   SELECT 1
   DO WHILE .NOT. EOF()
      n = 0
      mnday = nday
      DO WHILE  nday = mnday
         mnday = nday
         SKIP
         n = n + 1
      ENDDO
      SELECT 2
      REPLACE mon WITH mons[i]
      REPLACE inaug WITH LTRIM(STR(nyear))
      REPLACE F&mnday WITH n
      REPLACE dy WITH CDOW(CTOD("01/01/"+LTRIM(STR(nyear))))
      SELECT 1
   ENDDO
   i = i + 1
ENDDO

************ eop simp.prg


** this one fills in the table of years/days
USE bob EXCLU

SET SAFETY OFF
ZAP

cycle_span = 3000
ctr = 1
DO WHILE ctr <= cycle_span
   APPEND BLANK
   REPLACE yr WITH ctr - 1 + 2000
   ctr = ctr + 1
ENDDO
REPLACE ALL days WITH 364 FOR MOD(yr,5)   <> 0
REPLACE ALL days WITH 371 FOR MOD(yr,5)   = 0 .AND. MOD(yr,40) <>0
REPLACE ALL days WITH 364 FOR MOD(yr,40)  = 0 .AND. MOD(yr,400)<>0
REPLACE ALL days WITH 371 FOR MOD(yr,400) = 0

SUM days TO totaldays

? totaldays
************ eop bob.prg

*************************************************************************
** If you were to actually use this following program with the controlling
** program described above, you'd need to create the work file (temp8.dbf)
** and append from this one (call it temp8.prg) and make sure the line
** numbers match with what the controlling program expects (i.e., mjd =
** must be on line 30)

*************************************************************
* Program to generate Bob McClenon's Weekly Reformed Calendar
* coded by Alan Dechert
```

```
* these 2 tables are used:
*   Structure for database: BOB.DBF                              *
*   Field  Field Name  Type       Width    Dec    Index          *
*      1  YR          Numeric        7               N          *
*      2  DAYS        Numeric        5               N          *
*  ** Total **                      11                         *
*                                                               *
*   Structure for database: GDATES.DBF                          *
*   Field  Field Name  Type       Width    Dec    Index          *
*       1  JD          Numeric        8               N          *
*       2  GREGDATE    Date           8               N          *
*   ** Total **                      17                         *
****************************************************************
CLOSE DATABASES
SET TALK OFF
SET CENTURY OFF
SELECT 1
USE bob
SELECT 2
USE gdates
SELECT 1
daytrack        = 6
daycount        = 1
cycle_span      = 400
cumulative_days = 1
mjd             = 2455197

DECLARE ordmonth[13,2]
ordmonth[1,1]  = "January"
ordmonth[1,2]  = 31
ordmonth[2,1]  = "February"
ordmonth[2,2]  = 30
ordmonth[3,1]  = "March"
ordmonth[3,2]  = 30
ordmonth[4,1]  = "April"
ordmonth[4,2]  = 31
ordmonth[5,1]  = "May"
ordmonth[5,2]  = 30
ordmonth[6,1]  = "June"
ordmonth[6,2]  = 30
ordmonth[7,1]  = "July"
ordmonth[7,2]  = 31
ordmonth[8,1]  = "August"
ordmonth[8,2]  = 30
ordmonth[9,1]  = "September"
ordmonth[9,2]  = 30
ordmonth[10,1] = "October"
ordmonth[10,2] = 31
ordmonth[11,1] = "November"
ordmonth[11,2] = 30
ordmonth[12,1] = "December"
ordmonth[12,2] = 30
ordmonth[13,1] = "Leap Week"
ordmonth[13,2]  = 7

SET ALTERNATE TO temp1.txt
SET ALTERNATE ON

DO WHILE yr < cycle_span + 2010
   cntr = 1
   DO WHILE cntr <= 13
      IF cntr <> 13 .OR. days = 371
         ? ordmonth[cntr,1] + " " + LTRIM(TRANSFORM(yr,"999999"))   ;
```

```
         AT 23 - LEN(ordmonth[cntr,1] + " " +                        ;
         LTRIM(TRANSFORM(yr,"999999")))/2
         ? "Beginning with day "+ LTRIM(TRANSFORM(cumulative_days,;
         "999,999,999")) AT 23 - LEN("Beginning with day " +      ;
         LTRIM(TRANSFORM(cumulative_days,"999,999,999")))/2
         ? "Of the Reformed Weekly Calendar" AT 7
         ? "(Gregorian" AT 6
         SELECT 2
         GO mjd + cumulative_days - 2451544
         ?? gregdate AT 17
         SELECT 1
         ?? "JD" AT 26
         ?? mjd + cumulative_days PICTURE "9,999,999" AT 29
         ?? ")" AT 38
         ?
         DO PrintDays
         ?
         ?
      ENDIF
      cntr = cntr + 1
   ENDDO
   SKIP
ENDDO

SET ALTERNATE OFF
SET ALTERNATE TO

? "Done."

PROCEDURE PrintDays
   ? "      Sun  Mon  Tue  Wed  Thu  Fri  Sat "
   ?
   DO WHILE daycount <= ordmonth[cntr,2]
      ?? STR(daycount,4) AT daytrack * 5
      daytrack = IIF(daytrack = 7,1, daytrack + 1)
      IF daytrack = 1
         ?
      ENDIF
      daycount = daycount + 1
   ENDDO
   daycount = 1
   cumulative_days = cumulative_days + ordmonth[cntr,2]
RETURN
```