**SSRS Generate Calendar(s) based on a date range**

Updated calendar report based on time range show open Opportunity, MSA and SOW Count. Using recursive common-table-expressions I was able to generate the records needed to build a calendar for a given date range.

Here is the query:

--DECLARE @StartDate DATETIME = '03/26/2009', @EndDate DATETIME = '03/26/2010'

SELECT @StartDate = DATEADD(s,0,DATEADD(mm, DATEDIFF(m,0,@StartDate),0)) --FirstDayOfMonth

SELECT @EndDate = DATEADD(s,-1,DATEADD(mm, DATEDIFF(m,0,@EndDate)+1,0)) --LastDayOfMonth

; WITH Months AS (

SELECT

[Month] = DATEPART(MONTH,DATEADD(s,0,DATEADD(mm, DATEDIFF(m,0,@StartDate),0))),

[Year] = DATEPART(YEAR,DATEADD(s,0,DATEADD(mm, DATEDIFF(m,0,@StartDate),0))),

FirstDayOfMonth = DATEADD(s,0,DATEADD(mm, DATEDIFF(m,0,@StartDate),0)),

LastDayOfMonth = DATEADD(s,-1,DATEADD(mm, DATEDIFF(m,0,@StartDate)+1,0)),

FirstDayOfCalendar = DATEADD(DAY,-DATEPART(WEEKDAY,DATEADD(s,0,DATEADD(mm, DATEDIFF(m,0,@StartDate),0)))+1,DATEADD(s,0,DATEADD(mm, DATEDIFF(m,0,@StartDate),0))),

LastDayOfCalendar = DATEADD(DAY,6-DATEPART(WEEKDAY,DATEADD(s,-1,DATEADD(mm, DATEDIFF(m,0,@StartDate)+1,0))),DATEADD(s,-1,DATEADD(mm, DATEDIFF(m,0,@StartDate)+1,0)))

UNION ALL SELECT

[Month] = DATEPART(MONTH,DATEADD(MONTH,1,FirstDayOfMonth)),

[Year] = DATEPART(YEAR,DATEADD(MONTH,1,FirstDayOfMonth)),

FirstDayOfMonth = DATEADD(MONTH,1,FirstDayOfMonth),

LastDayOfMonth = DATEADD(s,-1,DATEADD(mm, DATEDIFF(m,0,DATEADD(MONTH,1,FirstDayOfMonth))+1,0)),

FirstDayOfCalendar = DATEADD(DAY,-DATEPART(WEEKDAY,DATEADD(MONTH,1,FirstDayOfMonth))+1,DATEADD(MONTH,1,FirstDayOfMonth)),

LastDayOfCalendar = DATEADD(DAY,6-DATEPART(WEEKDAY,DATEADD(s,-1,DATEADD(mm, DATEDIFF(m,0,DATEADD(MONTH,1,FirstDayOfMonth))+1,0))),DATEADD(s,-1,DATEADD(mm, DATEDIFF(m,0,DATEADD(MONTH,1,FirstDayOfMonth))+1,0)))

FROM

Months

WHERE

LastDayOfMonth < @EndDate

), Dates AS (

SELECT

[Month],

[Year],

[Date] = FirstDayOfCalendar,

FilterDate = LastDayOfCalendar

FROM

Months

UNION ALL SELECT

[Month],

[Year],

[Date] = DATEADD(DAY,1,[Date]),

FilterDate

FROM

Dates

WHERE

[Date] < FilterDate

), EmployeeBirthDay AS (

SELECT

e.BirthDate,

[Day] = DATEPART(DAY, e.BirthDate),

[Month] = DATEPART(MONTH, e.BirthDate),

[Year] = DATEPART(YEAR, e.BirthDate),

FullName = e.FirstName + ' ' + e.LastName

FROM

DimEmployee e

) SELECT

DisplayOnCalendar = DENSE\_RANK() OVER (ORDER BY d.Year, d.Month),

d.Month,

[Day] = DATEPART(DAY,d.[Date]),

d.Year,

[WeekDay] = DATEPART(WEEKDAY, d.[Date]),

[Order] = DENSE\_RANK() OVER (PARTITION BY d.Year, d.Month ORDER BY d.Date),

d.Date,

ebd.FullName

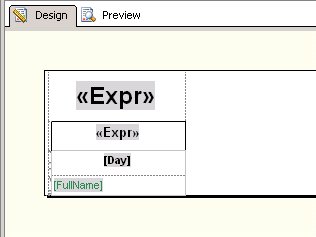
FROM

Dates d

LEFT JOIN EmployeeBirthDay ebd ON ebd.Month = DATEPART(MONTH,d.[Date]) AND ebd.Day = DATEPART(DAY,d.[Date])

OPTION (MAXRECURSION 1000)

Add a new tablix which is grouped by “DisplayOnCalendar” with page break after each group. Move the Month information from the report header into the new tablix as the group header. Drag the previous tablix into the details section on the new tablix. You can now remove the “Note” column from the previous tablix and update the value on the “Time” column to look at “FullName.” I made some additional format changes but won’t go in to details on those. Below is an example of what my report looks like before I preview it.



Once I preview the report I will show you the report which includes all the employees from AdventureWorks with their birthday in February 2010. The Date range I used is 03/26/2009 through 03/26/2010 so you will see there is 13 pages of the report to reflect the 13 months in the date range.



Here is a link to download the RDL file for the report above (<http://www.box.net/shared/o7i6pc2znt>) the zip file also contains the original RDL for the current month calendar report. The current month report is “Calendar.rdl” the updated version is “RangeCalendar.rdl” these reports will work with SSRS 2008 and the Adventureworks database.