**SSRS Expressions Tips and Tricks**

**By**[**Frank Fernando**](http://www.sqlservercentral.com/Authors/Articles/Frank_Fernando/390097/)**, 2009/12/21**

This article should come in handy for those that are new to SQL Server Reporting Services (SSRS) as well as a good reference for seasoned developers. I have included a list of expressions, custom functions and tips/tricks that have helped save me numerous hours over the years. It is not a comprehensive list, but some of the more commonly used ones in my experience.

A lot of these expressions/functions relates to formatting and extracting DATE/TIME, if you haven't already worked on SSRS you will soon find that you will be using them a lot. In fact I haven't had a report that doesn't use date/time and almost always one or more of these expressions/functions would come into play. Most expressions/functions covered in this article are built-in but it also covers some custom functions written in vb.net. The article also covers some properties/functionalities that are commonly used.

**I have broken down the article into 3 SECTIONS to make it a little easier to follow.**

1. The First section (EXPRESSIONS) covers the built-in expressions/functions available with SSRS. For most these expressions should cover all your basic requirements.
2. The second section (CUSTOM CODE/FUNCTIONS) looks at situations when the inbuilt expressions don't suffice. These custom code snippets should cover some of these scenarios. I have had to write these functions for my requirements over the years. These are custom vb.net functions that you can add and reference in your report.
3. The third Section (TIPS & TRICKS) should cover some of the properties and functionality that is available in SSRS but not used as often. Things like displaying report sin a single page, formatting rows dynamically etc.

***EXPRESSIONS***

A lot of these expressions would come in very handy when you setup report parameters. I would also suggest reading up on DateAdd, DatePart and other functions used in these expressions for more information. Note: all code are highlighted in blue.

1. Lets start with the easiest of the expressions i.e. To get Today’s date   
=Today

2. Get Date +/- a number of days - The following code will get Today's date -3 days. e.g. If today is the 08/12/2009, it should return 05/12/2009. You can change the -3 to any number and you should get the current date +/- the number specified. The DateAdd function basically adds an interval to a date. Here we add the interval -3 to Today's date.  
**=DateAdd("d", -3, Today)**

3. Get First Day of the Month - This is an extension of the pervious expression, it can be divided into two parts the -1.0 \* DatePart("D", Today) + 1 returns the number of days from the first day of the month -1 in negative. i.e. If today is the 08/12/2009 it would return the value -7 which is then passed to the function DateAdd to get the first day of the month e.g. 01/12/2009  
=DateAdd("D", -1.0 \* DatePart("D", Today) + 1, Today)

4. Get Last Day of the Month - This is almost the same as the last expression, the only difference is we don't add the +1 which gives us the last day of the last month then pass that date to another DateAdd function to add an extra month.  
=DateAdd("m", 1, DateAdd("d", -1.0 \* DatePart("d", Today), Today))

5. Get the First Day of the Last Month - This is another variation of the above expressions  
=DateAdd("D", -1.0 \* DatePart("D", Today) + 1, DateAdd("m", -1, Today))

6. Get the Name of the day - This gets the name for the day that was passed for example if its 08/12/2009 it would display "Tuesday"  
=WeekdayName(DatePart("w", Today))

7. Check what day it is and display date accordingly (This checks to see if today is a Monday and if it is true then displays today -3 days else displays today -1 day)  
=IIF(WeekdayName(DatePart("w", Today))="Monday",DateAdd("d", -3, Today),DateAdd("d", -1, Today))

8. Formatting Dates

|  |  |  |
| --- | --- | --- |
| Get Year - Will display only the year | **=Format(Today,"yyyy")** | Years as 1900-9999 |
| Get Month (without 0 suffix) | **=Format(Today,"M")** | Months as 1-12 |
| Get Month (with 0 suffix) | **=Format(Today,"MM")** | Months as 01-12 |
| Get Month by short Name | **=Format(Today,"MMM")** | Months as Jan-Dec |
| Get Month by Name | **=Format(Today,"MMMM")** | Months as January-December |
| Get Date (without 0 suffix) | **=Format(Today,"d")** | Days as 1-31 |
| Get Date (with 0 suffix) | **=Format(Today,"dd")** | Days as 01-31 |
| Get Date by short Name | **=Format(Today,"ddd")** | Days as Sun-Sat |
| Get Date by Name | **=Format(Today,"dddd")** | Days as Sunday-Saturday |
| Get Date in dd/mm/yyyy hh:mm:ss tt | **=Format(cdate("01/31/2009 16:00:00"),"dd MM yyyy hh:mm:ss tt")** | 31/01/2009 08:00:00 PM |

9. Convert String to Date - Following example combines string and integer values to form a valid date format  
=CDate( "1/" & Parameters!StartMonth.Value & "/" & Parameters!StartYear.Value)

10. SWITCH statement - An alternative to IIF/CASE (Returns a value depending on which condition is true)  
=SWITCH(WeekdayName(Fields!Date.Value) = "Monday","Blue",  
WeekdayName(Fields!Date.Value) = "Tuesday","Green",  
WeekdayName(Fields!Date.Value) = "Wednesday","Red")

11. Format Numbers as Currency - The result for the following will be $1000.00  
=FormatCurrency(1000)

12. Convert integer values to string  
= CStr(123123)

***CUSTOM CODE FUNCTIONS***

You will find that not all your requirements can be easily accomplished with functions available in SSRS. To accomplish this we can use the custom code option to write our own vb.net functions in the report and then reference them from the report. You will have to copy this code to the code section of the report properties (Click on the Report Menu and select Report Properties, in the dialog box that pops up select the Code tab and paste these functions)

1. Get Time in hh:mm:ss from seconds - In scenarios where you get the output from the database in seconds but you want the results to be displayed in the format hh:mm:ss you can use this function to pass the seconds value to the function and it will return a string value in the format of hh:mm:ss. Sample Usage of the Function from the report =***Code.GetDurationAsString(10000)***. This should return 02:46:40

Insert the code below into the Code Section under the Report Properties.

Code Description: The function below accepts values in double and returns value as string. We declare 3 variables one each to store the hours, minutes and seconds values. Then use the following formula to get the hours Math.Floor(total / 3600) and then use the following formula Math.Floor((total - (hour \* 3600)) / 60) to get the minutes and finally use the following Math.Floor(total - (hour \* 3600) - (min \* 60)) for the seconds. We then declare a variable retThis to store the string value that we would be returned, then put the hours, mins and second values together and pad them with 0 accordingly and store it in the retThis variable and finally return the calculated retThis value

Function GetDurationAsString(ByVal total As Double) As String  
Dim hour As Integer  
hour = Math.Floor(total / 3600)  
Dim min As Integer  
min = Math.Floor((total - (hour \* 3600)) / 60)  
Dim sec As Integer  
sec = Math.Floor(total - (hour \* 3600) - (min \* 60))  
Dim retThis As String  
If hour > 999 Then  
retThis = Microsoft.VisualBasic.Right("0000" + RTrim(CType(hour, String)), 4) & ":" & Microsoft.VisualBasic.Right("00" & RTrim(CType(min, String)), 2) & ":" & Microsoft.VisualBasic.Right("00" & RTrim(CType(sec, String)), 2)  
ElseIf hour > 99 Then  
retThis = Microsoft.VisualBasic.Right("000" + RTrim(CType(hour, String)), 3) & ":" & Microsoft.VisualBasic.Right("00" + RTrim(CType(min, String)), 2) & ":" & Microsoft.VisualBasic.Right("00" & RTrim(CType(sec, String)), 2)  
Else  
retThis = Microsoft.VisualBasic.Right("00" + RTrim(CType(hour, String)), 2) & ":" & Microsoft.VisualBasic.Right("00" + RTrim(CType(min, String)), 2) & ":" & Microsoft.VisualBasic.Right("00" & RTrim(CType(sec, String)), 2)  
End If  
Return retThis  
End Function

2. Get time in seconds from hh:mm:ss - This function is the reverse of the previous function where you pass the string values in hh:mm:ss and it will return the values in seconds. Sample Usage of the Function from the report=Code.GetTimeinSec(12:40:45)

Insert the code in blue below into the Code Section under the Report Properties.

Code Description: The function GetTimeinSec accepts string values in hh:mm:ss format. This value is then broken up into 3 parts i.e. hours,mins and seconds using the Substring function. Each of the part is converted to integer using the Convert.ToInt32 function. We then calculate the seconds of each of the part i.e. hh \* 360 to get the seconds from the hour, mm \* 60 to get the seconds from the minutes and finally the ss for the seconds. All the calculated parts are then added up together and returned back.

Function GetTimeinSec(ByVal textVal As String) As Integer  
Dim Total As String = ""  
Total = Convert.ToInt32(textVal.Substring(0, 2)) \* 360 + Convert.ToInt32(textVal.Substring(3, 2)) \* 60 + Convert.ToInt32(textVal.Substring(6, 2))  
Return Convert.ToInt32(Total)  
End Function

3. Check for Divide by Zero - You will find that when setting up reports especially with calculated columns in tables there a situations where you get a divide by zero error . You can use this function to make sure that SSRS doesn't display the ugly #error. Sample Usage of the Function from the report:

=Code.isDividebyZero(1,0))

Insert the code in blue below into the Code Section under the Report Properties.

Code Description: The Function IsDividedbyZero accepts two values, the numerator and the denominator. The function checks to see if the denominator is 0 and if it is it returns a 0 else it returns the numerator / denominator value. This functions does the division and also checks if there is a divide by zero error.

Function IsDividedbyZero(ByVal Numerator As Double, ByVal Denominator As Double) As Double  
If Denominator = 0 Then  
IsDividedbyZero = 0  
Else  
IsDividedbyZero = Numerator / Denominator  
End If  
Return IsDividedbyZero  
End Function

4. Open a link in a new window and control the window properties - SSRS does not have an built-in option to open a link in a new browser window and neither does it have any option to control the size and properties of the window. This function uses a JavaScript workaround to achieve this functionality.

Sample Usage:=Code.NewWindows ("www.google.com.au","no","no","no"))

This will open a new window that opens the google website with the toolbars, menubar and statusbar hidden.

Insert the code in italics blue into the Code Section under the Report Properties.

Code Description: The function NewWindows accepts 4 parameters, the url, toolbars, menubar and statusbar. The url parameter is the url you want to open, the statusbar, menubar and statusbar accepts a value of yes, no or a blank value (if its blank its considered as yes). The values passed are combined with a javascript (window.open) function and returned back. Lookup javascript window.open on google to get more information. You can easily modify this function to add more functionality to it eg. window size etc.

Public Function NewWindows(ByVal url As String, ByVal toolbars As String, ByVal menubar As String, ByVal statusbar As String) As String  
If toolbars IsNothing Or toolbars = "" Then  
toolbars = "yes"  
End If  
If menubar IsNothing Or menubar = "" Then  
menubar = "yes"  
End If  
If statusbar IsNothing Or statusbar = "" Then  
statusbar = "yes"  
End If  
Return "javascript:void(window.open('http://" & url & "','\_blank','toolbar=" \_  
& toolbars & ",menubar=" & menubar & ",statusbar=" & statusbar & "'))"  
End Function

5. Display Popup Message on Click - This function gives you an option to display a message box from within the report items. For example you might want a popup when a user click on an image. You just add the following to the Jump to URL expression in image properties (right click the image and select properties - Select the Navigation Tab and select the radio button Jump to URL and copy the following). You can customize the message by passing report values. This opens a message box with the text "Please Contact Support on help@support.com".  
="javascript:void(alert('Please Contact Support on help@support.com'))"

***TIPS & TRICKS***

I have included this third category as these dont quiet fit into the other 2 sections. This section covers functionality that's available in SSRS but might not be used as often or users just dont know about it.

1. Color alternate rows in table with a different color - This is very useful when there are a lot of rows with similar data and its hard to differentiate the rows. You have to set the table background color property with the following code.  
=iif(RowNumber(Nothing) Mod 2, "WhiteSmoke", "White")

2. To disable pagination - If you want to display the report in one single page you have to change change the report properties InteractiveHeight and InteractiveWidth to 0

3. Passing parameters to SSRS through the URL - If you plan to link to a report from your own website and if the report accepts parameters you can use the following sample to open the report pass the parameter values from the addressbar. This example passes 4 parameters Centre\_Contract, Emp\_ID,Start\_Date,End\_Date. Parameters start with rs:Command and each of the parameters are separated using & ).   
http://reporting.acme/ReportServer/Pages/ReportViewer.aspx?%2fUPA+Reports%2fUPASummary&rs:Command=Render&Centre\_Contract=21&Emp\_ID=11170&Start\_Date=2009-07-05&End\_Date=2009-07-06

4. Hide the parameters in a report - This is the same as the last option but if you add the following command (&rc:Parameters=false) at the end of the report url it will hide the parameters in the report. Very useful when you want to control what data can be viewed by which person especially from 3rd party applications.  
http://reporting.stellar/ReportServer/Pages/ReportViewer.aspx?%2fUPA+Reports%2fUPASummary&rs:Command=Render&Centre\_Contract=21&Emp\_ID=11170&Start\_Date=2009-07-05&End\_Date=2009-07-06&rc:Parameters=false

5. Dynamic DataSource - If you have a requirement for a report to be run against multiple database server you can change the datasource connection string property to be dynamic. You can write your own expression in the connection string option in the data source properties. The following will create a dynamic connection to the server depending on the report parameter (ServerName) selected. Note: You have to make sure that the login has permission on all the servers you would like to connect to.

="Data Source=" & Parameters!ServerName.Value & ";Initial Catalog=WSS\_Content"

6. As a quick note, did you know that you can upload Excel, Access, HTML and other files to reporting services. Very handy especially if you want to include a help file with your report etc

Hope these expressions/functions/tips have come in handy for you. I am currently about half way through in writing up a complete class with many more functions and should have them uploaded in a month or so.