

RB_Text_Format - Format Text



Function

This function will return a string variable passed as parameter 1 formatted to print with a line length of parameter 2. It will break each line at the end of a word

This routine is useful for printing a memo, or other text field, on a report.



Calling Procedure

```
newstring = RB_Text_Format(oldstring, 65)
Printer.Print newstring
```

This will print the contents of oldstring as 65 character lines.



Notes

This routine will work with printer ScaleMode set to character or inches. The second parameter must correspond with the Printer ScaleMode currently set.

[Example Code](#)

[Files To Include To Use RDBLIB](#)

RB_Text_Format Sample Code

```
oldstring = SampleTable!Comments  
newstring = RB_Text_Format(oldstring, 65)  
Printer.Print newstring
```


RB_Validate_Date Function

Function

This function validates the input (date) contained in control passed as parameter 1. It will return True if the input is a valid date, the string "_/_/_" or null. It will display a msgbox with an "Enter a valid date" msg and return False if the input date is invalid.

Calling Procedure

TxtDate_LostFocus

```
IF Not RB_Validate_Date(TxtDate) then  
    TxtDate.setFocus  
End If
```

Notes

The input date can be entered in 1 or 2 digit mm/dd/yy format or as a 6 digit number. All of the following are valid formats for April 1, 1995: 4/1/95, 04/01/95, 04-01-95, 4-1-95, and 040195.

[Sample Code](#)

[Files To Include To Use RDBLIB](#)

RB_Validate_Date Sample Code

```
IF Not RB_Validate_Date(TxtDate) then  
    TxtDate.setfocus  
End If
```

RDBLIB Routines

RDBLIB is a set of common subroutines and functions for use in Visual Basic Programs. The followings functions are available:

[RB_ErrorHandler - Error Handling Routine](#)

[RB_Center - Center String On Print Output](#)

[RB_RJustify - Right Jusify Numeric Value](#)

[RB_StatusMsg - Obtain Status Msg From Tag Property](#)

[RB_Text_Format - Format Text String For Printing](#)

[RB_Validate_Date - Date Validation Routine](#)

[ShellAndWait - Shell Program And Wait For It To Terminate](#)

[Performance Logging Functions](#)

[Files To Include To Use RDBLIB](#)

RB_ErrorHandler

Function

This sub-routine provides standardized error handling for a Visual Basic program. It will display a message box indicating the error that occurred and provide options to create a problem report, display help on the error, ignore the error, retry the function, or abort the program. The program will also log the error to the logging database (RDBLOG.RDB). Queries are available in it to display the errors.

The function returns a code indicating the user's choice to retry or ignore the error.

Calling Procedure

```
erraction = RB_ErrorHandler("FormName", "Routine")
Select Case erraction
Case 1
    Resume 0      ' Retry option selected
Case 2
    Resume Next   ' Ignore option selected
End Select
```

FormName is the name of the form where the error occurs

Routine is the name of the routine where the error occurs. You can provide as much information in the EventName parameter as is needed to help you identify where the error occurs. As a minimum it should be the event name.

The value returned (erraction above) indicates the users choice for handling the error. A value of 0 indicates the Retry choice, 1 indicates the Ignore choice. The Abort option terminates the program before returning to the calling routine.

Prior to calling the error routine, normally in the main form load event, you can set the following two variables to automatically fill in on the problem report.

```
RB_systemname (As String) - the name of your system (program)
RB_version (As String) - the version number of your system (program)
```

Notes

These routines provide a means for the user to create a problem report that contains information that can assist the developer in determining the cause of the error condition. This report can be printed.

[Sample Code](#)

[Error Message Box](#)

[Problem Report Form](#)

[Files To Include To Use RDBLIB](#)

Files To Include To Use RDBLIB

RDBLIB.BAS
RBERRFRM.FRM
RBLOGOPT.FRM
RBPROBRP.FRM
RBSCRN.FRM

RB_Center - Centering Text



Function

This sub-routine will center a string passed as parameter 1 on the printer line passed as parameter 2 or current line if parameter 2 = 0. It Will skip to next line if parameter 3 = true



Calling Procedure

RB_Center (str_to_print As String, line_no, skip_line As Integer)

str_to_print = the string that is to print centered and printed

line_no = the line number to print the string on, if this value is zero the text will be printed on the current line

skip_line = true causes the routine to skip to the next line after printing the text, other values leave the current print line on the line specified by parameter 2

[Sample Code](#)

[Files To Include To Use RDBLIB](#)

```
RB_systemname = "RDBLIB Demo"  
RB_version = "1.1"
```

```
on error goto errorroutine
```

```
....  
....
```

```
errorroutine:
```

```
erraction = RB_ErrorHandler("MainForm", "Sample Code")
```

```
Select Case erraction
```

```
Case 1
```

```
Resume 0      ' Retry option selected
```

```
Case 2
```

```
Resume Next   ' Ignore option selected
```

```
End Select
```

RB_Center Sample Code

```
sString = "This is a string to centered by itself on line 6 of a printout"
```

```
RB_Center sString, 6, True
```

RB_RJustify - Right Justify Number To Specified Position

Function

This function will print a number passed as parameter 1 according to the format passed as parameter 2 right justified on the column passed as parameter 3. It returns the leftmost column position where printing started

Calling Procedure

```
leftcol = RB_Rjustify(200, "###,###.##", 40)
```

This will print " 200.00" with the rightmost 0 at column 40

Notes

This routine will work with the printer scalemode set to inches or characters. The third parameter must be appropriate for the current printer scale mode.

[Sample Code](#)

[Files To Include to Use RDBLIB](#)

RB_Rjustifly Sample Code

```
leftcol = RB_Rjustifly(200, "###,###.##", 40)
```

RB_StatusMsg - Return Status Bar Message

Function

This function can be used with VSVBX or other controls that use the first part of the tag property for some purpose and you want to use the second part of the tag property for your purposes. For example VSVBX can use the first part of the tag property for a label. I use the second part for a status bar message (hence the name StatusMsg). The separator that is used to indicate the separation of the parts is the pipe symbol (|).

Calling Procedure

Set the tag property of a control, e.g. 'Name|Enter the persons name'

```
StatusBar.Caption = RB_StatusMsg((TxtName.Tag))
```

This will display "Enter the persons name" in the status bar.

Notes

If you are passing the tag property directly, as in the example above, you must enclose it in parenthesis because the parameter is specified As Text.

[Sample Code](#)

[Files To Include To Use RDBLIB](#)

RB_StatusMsg Sample Code

```
StatusBar.Caption = RB_StatusMsg((TxtName.Tag))
```

ShellAndWait Subroutine



Function

This subroutine will start (via the Shell Function) the command passed as parameter 1 and wait until the command has completed and the window closed

This routine is useful if you do not want to continue execution until the program you started finishes.



Calling Procedure

```
ShellAndWait("COPY A.TXT B.TXT")  
' The B.TXT file will be available now
```



Notes

If you use the normal VB Shell function your program continues as soon as the other program is initiated.

The program you start can be a Windows or DOS program.

[Sample Code](#)

[Files To Include To Use RDBLIB](#)

ShellAndWait Sample Code

```
ShellAndWait("COPY A.TXT B.TXT")
```

Performance Logging Functions

RDBLIB provides a facility for logging the performance (response times) of components of your application. The basic steps required are:

[RB_SetLogOptions - Set logging options](#)

[RB_OpenLog - Open the log database](#)

[RB_LogTask - Log start and end of task](#)

[RB_CloseLog - Close the log database](#)

[Files To Include To Use RDBLIB](#)

There are a number of Queries defined in the logging database (RDBLOG.MDB) that can be used to display and analyze the performance data.

This facility can also be handy for troubleshooting at times. You can use the log records to determine where you were in your program when an error occurred.

RB_SetLogOptions

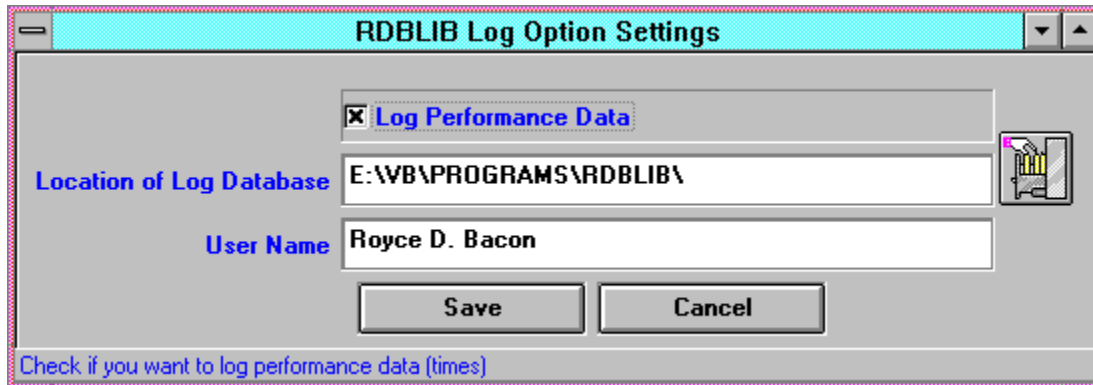
Function

This sub-routine can be called to set the performance and error logging options. It displays a form that allows you to indicate if you want performance data logged, where the logging database is located and the users name.

Calling Procedures

RB_SetLogOptions

Form



The screenshot shows a dialog box titled "RDBLIB Log Option Settings". It contains the following elements:

- A checked checkbox labeled "Log Performance Data".
- A text field labeled "Location of Log Database" containing the path "E:\WB\PROGRAMS\RDBLIB\".
- A text field labeled "User Name" containing the name "Royce D. Bacon".
- Two buttons: "Save" and "Cancel".
- A footer note: "Check if you want to log performance data (times)".

Check this box if you want to log performance data. If this box is not checked then performance data will not be logged.

Enter the full path for where the logging database (RDBLOG.MDB) is located. If you don't know the location use the browse button to the right of this field to locate the database.

Click on this button to obtain a Windows File browse dialog for locating the logging database.

Enter your name here. This name will appear in the error and performance log data to identify where the data came from.

Click this button to save the changes you have made into the RDBLIB.INI file.

Click this button to cancel the changes you have made.

RB_OpenLog Routine



Function

This routine is called to open the logging database. You should do this at the beginning of your program. Along with opening the logging database this routine will determine from the RDBLIB.INI file if you are logging performance data or not.



Calling Procedure

RB_OpenLog



Notes

If the routine can not open the logging database for some reason it will set the flag for logging performance data off.

RB_LogTask Function

Function

This routine is called to log the start or end of a task. The difference between the start and end determine the duration (response time) for a task.

Calling Procedure

RB_LogTask sTask, sForm, iFunction, sComments

Where:

sTask = a string identifying the task being timed.

sForm = a string identifying the form where this task is being performed

iFunction = RB_STARTTASK to indicate the start of a task or RB_ENDTASK to indicate the end of a task

sComments = any additional information regarding the task

Notes

RB_STARTTASK and RB_ENDTASK are defined as Global Constants in RDBLIB.BAS.

The RB_STARTTASK will log the ending time for the previous task if a RB_ENDTASK was not done.

The comments for a task can be set with either the time of the STARTTASK or ENDTASK function. The comments are useful for saving information that may indicate the detail of the function being done and why a function may be taking a long time. For example, if you are timing the execution of an SQL statement, comments can contain the SQL statement.

[Sample Code](#)

```
RB_LogTask ("Opening DataBase", "MainForm", RB_STARTTASK, sDataBaseName)
Set dbSampleDB = OpenDatabase(sDataBaseName)
sSQL = "SELECT * FROM 'Sample Table' WHERE key = '12345'"
RB_LogTask("Selecting Data", "MainForm", RB_STARTTASK, sSQL)
Set tbSampleDS = dbSampleDB.OpenDynaset(sSQL)
RB_LogTask("Selecting Data", "MainForm", RB_ENDTASK, "")
```

RB_CloseLog Routine



Function

This routine is called to close the logging database when your program is ending.



Calling Procedure

RB_CloseLog

RDB Error Handler

A Duplicate definition error (10) has occurred at line 10 in routine BtnGenError_Click of form MainForm.

Press Help for more information

Abort

Retry

Ignore

Help

Create Problem Report

Problem Report Form

Problem Reported By Royce D. Bacon **On** 04/01/95 - 17:51:58

Organization..... RDB Systems **Contact Person** Same

System..... RDBLIB Demo **Version**..... 1.3

Type of Problem

- System Ended Abnormally System Didn't Work As Expected Problem is Repeatable (Consistent)
 Other (Please Specify) Sample Error Report

Effect Of Problem

- Can't Use System Can't use this function Can Use System, But Functionality Is Limited
 Other (Please Specify) Sample Error Report

Describe the function being performed, the actions leading up to the error and the problem that occurred

This is a sample error report. In this section the user would describe what they were doing leading up to the function that created the error.

Visual Basic Error Message A Duplicate definition error (10) has occurred at line 10 in routine BtnGenError_Click of form MainForm.

Select Printer Print Problem Report Exit

This message indicates the error that occurred.

Click this button to abort (end) the program now.

Click this button to retry the function that originally created the error.

Click this button to ignore the error and continue on to the next statement.

Click this button to obtain additional information about the cause of the error and potential resolutions to it.

