

# RDLC Report Custom Code Utility Functions

---

This module provides comprehensive utility functions for RDLC reports including string manipulation, number conversion to words with Indian currency format, global data management, and logging capabilities.

**Source:** Merged from [frontlook-admin/RDLCReport\\_CustomCode](#)

## Table of Contents

- [Logging Functions](#)
  - [Global Data Management](#)
  - [String Manipulation Functions](#)
  - [Number to Words Conversion](#)
- 

## Logging Functions

### WriteLog (Simple)

Writes a log message to a file with timestamp. Always generates a new path (stateless).

#### Parameters:

- **message** (String): The message to log
- **filePath** (String, Optional): Directory path for the log file. Default: "C:\Temp"
- **fileName** (String, Optional): Base name for the log file. Default: "CliReportDebug\_yyyyMMdd"

#### Behavior:

- If no filename is provided, creates: **CliReportDebug\_20251011.log**
- If filename is provided, creates: **CustomName\_20251011.log**
- Automatically creates the directory if it doesn't exist
- Each log entry is timestamped with format: **yyyy-MM-dd HH:mm:ss.fff**
- Logging errors are silently ignored

```
' Default usage - logs to C:\Temp\CliReportDebug_20251011.log
WriteLog("This is a test message")

' Custom filepath - logs to D:\Logs\CliReportDebug_20251011.log
WriteLog("Custom path message", "D:\Logs")

' Custom filepath and filename - logs to D:\Logs\MyReport_20251011.log
WriteLog("Custom file message", "D:\Logs", "MyReport")
```

### WriteLogCached (Performance)

Cached version of WriteLog for better performance with multiple log entries in same session.

#### Benefits:

- Caches log file path for repeated writes
- Automatically detects date changes and updates path
- Ideal for multiple log writes in same report execution

```
' Multiple logs in same session - more efficient
WriteLogCached("Starting report processing")
WriteLogCached("Processing item 1")
WriteLogCached("Processing item 2")
WriteLogCached("Report completed")
```

---

## Global Data Management

Transfer data from report body to headers/footers using named key-value pairs.

### SetGlobalData

Sets global data from a key-value list. Call this in a hidden tablix cell.

#### Usage in RDLC:

```
=Code.SetGlobalData(Fields!GlobalData.Value)
```

#### In C/AL or AL (to create the key-value list):

```
local procedure AddKeyValue(VAR KeyValueListAsText: Text; _Key: Text; _Value:
Text)
var
    Chr177: Text[1];
    NewPair: Text;
begin
    Chr177[1] := 177;
    NewPair := _Key + Chr177 + _Value + Chr177;
    KeyValueListAsText += NewPair;
end;

local procedure GetGlobalDataFields() KeyValueList : Text
begin
    AddKeyValue(KeyValueList, 'CompanyName', CompanyInfo.Name);
    AddKeyValue(KeyValueList, 'CompanyAddress', CompanyInfo.Address);
    AddKeyValue(KeyValueList, 'ReportDate', Format(Today));
end;
```

## GetVal

Retrieves a value from global data by name or index.

### Parameters:

- **Key** (String or Number): The key name (case-insensitive) or numeric index (1-based)

### Returns:

- The value, or error message if not found (e.g., **?KeyName?**)

### Usage in RDLC:

```
=Code.GetVal("CompanyName") '  
=Code.GetVal("ReportDate") '  
=Code.GetVal(1) '
```

**Note:** End expressions with an apostrophe (') to preserve arguments when copy/pasting textboxes.

## AddKeyValue

Adds or updates a key-value pair in a collection.

### Parameters:

- **Data** (Collection): The collection to modify
- **Key** (String): The key (case-insensitive)
- **Value** (Object): The value to store

---

## Legacy NAV Way (SetData & GetData)

For backward compatibility with traditional NAV reports using numbered data groups.

### SetData

Sets data in one of three global variables (Data1, Data2, or Data3).

### Parameters:

- **NewData** (String): String with Chr(177) as separator
- **Group** (Integer): Which global variable to use (1, 2, or 3)

### Usage:

```
=Code.SetData(Fields!GlobalData.Value, 1)  
=Code.SetData(Fields!HeaderData.Value, 2)
```

## GetData

Gets data by position number from one of three global variables.

### Parameters:

- **Num** (Integer): Position number of the value (1-based)
- **Group** (Integer): Which global variable to use (1, 2, or 3)

### Usage:

```
=Code.GetData(1, 1) ' Gets first value from Data1  
=Code.GetData(3, 2) ' Gets third value from Data2
```

**Note:** The improved SetGlobalData/GetVal approach is recommended over SetData/GetData as it provides:

- Named keys instead of position numbers
- Better readability (`GetVal("CompanyName")` vs `GetData(5, 1)`)
- Single global collection instead of three separate variables
- Case-insensitive key access

---

## String Manipulation Functions

These functions help concatenate strings with various separators, filtering out empty values.

### ConcatenateNonEmptyWithCrLf

Concatenates non-empty strings from an array with CRLF (new line) characters.

```
Dim result As String = ConcatenateNonEmptyWithCrLf(New String() {"Hello", "",  
"World"})  
' Result: "Hello<CRLF>World"
```

### ConcatenateNonEmptyWithCrLfAndDelimiter

Concatenates non-empty strings with the specified delimiter.

```
Dim result As String = ConcatenateNonEmptyWithCrLfAndDelimiter(New String()  
{"Hello", "", "World"}, ",")  
' Result: "Hello,World"
```

### ConcatenateWithCrLf

Joins all strings with CRLF (new line) characters, including empty strings.

```
Dim result As String = ConcatenateWithCrLf(New String() {"Hello", "", "World"})  
' Result: "Hello<CRLF><CRLF>World"
```

## Number to Words Conversion

### ToWordsIn (Double)

Converts a numeric value to its word representation in Indian format with optional currency formatting.

```
Dim result As String = ToWordsIn(1234.56, True, True)  
' Result: "Rupees One Thousand Two Hundred Thirty-Four And Fifty-Six Paise  
Only"
```

### ToWordsIn (Long)

Converts a Long integer to its word representation using the Indian numbering system.

```
Dim result As String = ToWordsIn(1234567)  
' Result: "Twelve Lakh Thirty-Four Thousand Five Hundred and Sixty-Seven"
```

### FL\_NumberToWordsMinimised

Creates a shorter representation of numbers using appropriate Indian units.

```
Dim result As String = FL_NumberToWordsMinimised(150000)  
' Result: "1.5 Lakh"
```

## Complete Usage Example

### In C/AL or AL Code

```
local procedure AddKeyValue(VAR KeyValueListAsText: Text; _Key: Text; _Value:  
Text)  
var  
    Chr177: Text[1];  
    NewPair: Text;  
begin  
    Chr177[1] := 177;  
    NewPair := _Key + Chr177 + _Value + Chr177;  
    KeyValueListAsText += NewPair;
```

```

end;

local procedure GetGlobalDataFields() KeyValueCollection : Text
begin
    AddKeyValue(KeyValueCollection, 'CompanyName', CompanyInfo.Name);
    AddKeyValue(KeyValueCollection, 'Address', CompanyInfo.Address);
    AddKeyValue(KeyValueCollection, 'ReportDate', Format(Today));
end;

```

## In RDLC Report

```

' Hidden tablix cell to set global data
=Code.SetGlobalData(Fields!GlobalData.Value)

' Header/Footer - Get values by name
=Code.GetVal("CompanyName")
=Code.GetVal("Address")
=Code.GetVal("ReportDate")

' Logging
=Code.WriteLog("Report generated for: " & Fields!CustomerName.Value)

' String concatenation
=Code.ConcatenateNonEmptyWithCrLf(New String() {Fields!Line1.Value,
Fields!Line2.Value})

' Number to words
=Code.ToWordsIn(Fields!TotalAmount.Value)

```

## Files

- **RdlcReportCode.vb** - Main code file (copy to RDLC Custom Code section)
- **RdlcReportCode\_WithComments.vb** - Fully documented version with XML comments
- **RdlcVBCode\_Usage** - Usage examples for all functions
- **Readme.md** - This documentation file

## Credits

- Global Data Management functions from: [frontlook-admin/RDLCReport\\_CustomCode](https://github.com/frontlook-admin/RDLCReport_CustomCode)
- Original concept by Andreas Rascher: [AndreasRascher/RDLCReport\\_CustomCode](https://github.com/AndreasRascher/RDLCReport_CustomCode)