

"Tip of the Day" Digest





"Tip of the Day" Digest

by R:BASE Technologies, Inc.

Welcome to the "Tip of the Day" Digest

The "Tip of the Day" Digest is a collection of the "Tip of the Day" threads posted on R:BASE List Servers from Razzak Memon, an R:BASE developer himself, the President & CEO of R:BASE Technologies, Inc.

These tips include many hundreds of features and innovations that have been added to R:BASE over the years.

"Tip of the Day" Digest

Copyright © 1982-2016 R:BASE Technologies, Inc.

Information in this document, including URL and other Internet web site references, is subject to change without notice. The example companies, individuals, products, organizations and events depicted herein are completely fictitious. Any similarity to a company, individual, product, organization or event is completely unintentional. R:BASE Technologies, Inc. shall not be liable for errors contained herein or for incidental consequential damages in connection with the furnishing, performance, or use of this material. This document contains proprietary information, which is protected by copyright. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written consent of R:BASE Technologies, Inc. We reserve the right to make changes from time to time in the contents hereof without obligation to notify any person of such revision or changes. We also reserve the right to change the specification without notice and may therefore not coincide with the contents of this document. The manufacturer assumes no responsibilities with regard to the performance or use of third party products.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

The software described in this document is furnished under a license agreement. The software may be used or copied only in accordance with the terms of that agreement. Any unauthorized use or duplication of the software is forbidden.

R:BASE Technologies, Inc. may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from R:BASE Technologies, Inc., the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Trademarks

R:BASE®, Oterro®, RAdmin®, R:Scope®, R:WEB Suite®, R:Mail®, R:Charts®, R:Spell Checker®, R:Docs®, R:BASE Editor®, R:Scheduler®, R:BASE Plugin Power Pack®, R:Style®, R:Code®, R:Struc®, RBZip®, R:Fax®, R:QBDataDirect®, R:QBSynchronizer®, R:QBDBExtractor®, R:Mail Editor®, R:Linux®, R:BASE Dependency Viewer®, R:Archive®, R:Chat®, RDCC Client®, R:Mail Editor®, R:Code®, R:Column Analyzer®, R:DF Form Filler®, R:FTPClient®, R:SFTPClient®, RBMap®, R:GeoCoder®, R:PDF Form Filler®, R:PDFWorks®, R:PDFMerge®, R:PDFSearch®, RBInstaller®, RBUpdater®, R:Capture®, R:RemoteControl®, R:Synchronizer®, R:Biometric®, R:CAD Viewer®, R:DXF®, R:Twain2PDF®, R:Tango®, R:SureShip®, R:BASE Total Backup®, R:Scribbler®, R:SmartSig®, R:OutLink®, R:JobTrack®, R:TimeTrack®, R:Syntax®, R:WatchDog®, R:Manufacturing®, R:Merge®, R:Documenter®, R:Magellan®, R:WEB Reports®, R:WEB Gateway®, R:Stat®, R:ReadyRoute®, R:Accounting®, R:Contact®, R:DWF Viewer®, R:Mail Viewer®, R:Java®, R:PHP® and Pocket R:BASE® are trademarks or registered trademarks of R:BASE Technologies, Inc. All Rights Reserved. All other brand, product names, company names and logos are trademarks or registered trademarks of their respective companies.

Windows, Windows 10, Windows 8.x, Windows 7, Vista, Windows Server 2003-2012 R2, XP, Bing Maps, MapPoint, and Outlook are registered trademarks of Microsoft Corporation.

Printed: December 2016 in Murrysville, PA

Fifth Edition

Table of Contents

Part I General	2
1 Capturing Computed Column Expressions	2
2 Configuring Anti-Virus Programs to Work with R:BASE	3
3 Copying Forms, Labels, and Reports	4
4 Customized Input Language for the ROSK	4
5 Customizing the R:BASE On-Screen Keyboard (ROSK)	5
6 Customizing ROSK for R:BASE Mobile Applications	5
7 Data Dictionary and Selecting Color Value	6
8 Defining a Default Path for "Open Image" Dialogs	7
9 Deploying an R:BASE X Enterprise Database	7
10 Display the Selected Expression Builder Variable	8
11 Displaying R:BASE Keys/Indexes	8
12 Documenting Component IDs in Forms/Labels/Reports	9
13 Dynamically Changing Themes in R:BASE Application (.RBA)	10
14 Dynamically Defining Unique File Names	11
15 Enabling Object Inspector	12
16 External Print Module Within the TabPrint.dll	12
17 Finding the Exact Name of a Form Control Type	13
18 Handling Milliseconds in Parameterized Update Statements	14
19 How Healthy is your Database?	14
20 Implementing Application Privileges	16
21 Is the Application Compiled or Not?	16
22 Is the Application Runtime or Not?	16
23 Managing Custom Colors	17
24 Managing External Themes - R:BASE Themes Unleashed!	18
25 Managing Multipage Images	18
26 New Data Types in R:BASE X Enterprise	19
27 Optimizing the R:BASE Multi-User Environment	20
28 R:Backup Utility	22
29 R:BASE X or R:BASE X Enterprise?	23
30 The Complete R:BASE System Variable List	24
31 The R:BASE On-Screen Keyboard (ROSK)	27
32 Tracking Date/Time and Version Information	29
33 Update Command History	32
34 Using Expanded Color Palette	34

35	Using Extended Colors	35
36	Using Foreign Keys in TEMPORARY Tables	36
37	Using R:BASE Executable Startup Options	38
38	Using the #NOW System Variable	40
39	Using the Enhanced Find in Files Utility	41
40	Using the Magnifying Glass	41
41	Using the Scratch Pad	42
42	What Version of Engine DLL, and Files In Use?	42
Part II Settings		45
1	About the COLCHECK Setting	45
2	Assigning Default Component IDs for Form, Label, and Report Controls	45
3	AUTOMUMbering in a Multi-User Environment	45
4	Automating the Directory for SCRATCH Files	46
5	Cleaning Scratch (.\$\$\$) Files on Exit	46
6	Customizing Default Printer Settings	47
7	Customizing Hints	47
8	Customizing WALKMENU Time Interval Settings	49
9	Customizing the Forms/Reports/Label Expression Builder	49
10	Database Stored Settings	50
11	Disabling TRACE in Forms, Labels and Reports	51
12	Duplicating Your Personalized R:BASE Settings	52
13	Finding Default vs. Active SCRATCH Settings	53
14	Handling Form/Report/Label Designer -ERROR-Messages	54
15	New PAGELock Setting	54
16	Preserving Computed Column Values When Using UNLOAD/LOAD	55
17	Turning Off User Interface Updates to Increase Performance	56
18	Using Conditional TRACE in a Command File	56
19	Using Double-Click to Bring Up the Designers	57
20	Using the CHECKPROP Setting	57
21	Warning When Changing Table Structure	58
22	Warning When Closing an R:BASE Session	58
Part III R> Prompt		61
1	Copying/Pasting Multiple Line Code	61
2	Documenting and Recycling the Command History	61
3	Hot Keys	61
Part IV Database Explorer		64
1	Customizing the Database Explorer	64

2	Default File Names for Unloading Forms/Labels/Reports	65
3	Exploring the Database Explorer	66
4	Form Preview From the Database Explorer	68
5	Key Word Search Options	68
6	One Click UNLOAD for Data and Structure	69
7	Open Table/View Options Using the Data Browser/Editor	69
8	Renaming Database Explorer Objects	70
9	Resetting Version Number of External Form Files	70
10	Searching for Text in Control Properties	71
11	Searching for Text in Expressions	73
12	Searching in Database Explorer	74
13	Using MDI Option for Database Explorer Modules	75
14	Using the Find in Files Utility	77
Part V Data Designer (RBDefine)		80
1	Computed Columns and Default Values	80
2	Reorganizing Columns in Data Designer	80
3	Validate the syntax for an email address	80
Part VI Data Browser		83
1	Adding a Duplicate Row in the Data Browser/Editor	83
2	Conditionally Managing Data Browser/Editor Options	83
3	Data Browser Export Options	89
4	Right-Click Calculate (Compute, Tally, Duplicates)	89
Part VII Stored Procedures		91
1	Order of Processing for Stored Procedures and Triggers	91
Part VIII Forms		93
1	Accessing the Control List	93
2	Action Sequence (Form EEP Execution)	95
3	Added Advanced Variable Rich Edit Control	95
4	Additional Fields for DB/Variable Email Label Controls	95
5	Aligning Radio Groups Buttons	97
6	Assigning Keys to Execute Custom Form Actions	97
7	Automating Date/Time Fields	98
8	Capturing Column Name and Sort Order	98
9	Capturing the Value of Form Component IDs	99
10	Choosing the Right Form Control	100
11	Clearing the PDF Viewer Document	101
12	Click-Based RBTI System Variables	102

13	Controlling Date/Time Picker Drop-Down	102
14	Controlling TABLE Add/Delete Row Properties	102
15	Converting DB Grid Control to Enhanced DB Grid Control	103
16	Converting R:BASE Application Files to External Form Files	103
17	Converting Variable Form to External Form File	105
18	Copy and Paste Custom Form Actions	105
19	Copy and Paste Expression Builder Variables	105
20	Copy and Paste Menu Bar Items	106
21	Copy and Paste Tree View Nodes	106
22	Copying and Pasting Windows Clipboard Data in Forms	107
23	Creating a Dynamic Tree View Object	108
24	Customized Column Format in Variable Lookup List Views	109
25	Customizing Character Case for Data Input	110
26	Customizing DB Grid Column Properties	110
27	Customizing Filter Options for the Enhanced DB Grid Control	111
28	Customizing the Form Closing Behavior	112
29	Customizing the Form Menu Bar Appearance	113
30	Customizing the Selected Row Highlight Color in List Views	115
31	Customizing Lookup List View Column Headers/Details	115
32	Defining Custom EEPs for "On Roll Up" and "On Roll Down"	118
33	Defining TEMPORARY TABLES/VIEWS with Actions	119
34	Defining the Number of Items in Pop-up Menus	120
35	Designing Custom Pop-up Menus	120
36	Disabling the Sort Feature for DB Grid Columns	120
37	Displaying a Balloon Tip in DB/Variable Edit Fields	121
38	Displaying a Cue Banner in DB/Variable Edit Fields	121
39	Displaying a Row Number in List Views	123
40	Displaying the Current Table/Field on the Status Bar	124
41	Displaying the Record Number in a DB Grid	124
42	Displaying Thumbnail Images With the File List Box	124
43	Documenting Form EEPs	125
44	Dynamically Changing Band List Properties	126
45	Dynamically Passing DB Grid Filter Parameters	128
46	Easily Saving Table Image Data as a File	130
47	Enabling/Disabling the "Drag and Drop" Feature	130
48	Enhanced DB Grid Control - Hot Keys	131
49	Focusing on the Enhanced DB Grid Filter Bar	132
50	Getting the Radio Button Count	132
51	GROUPINDEX property for Speed Buttons	133

52	Listing All Controls with Component IDs	133
53	Loading Images in Drop-Down and Bit Buttons	134
54	Making an Enhanced Group Box Transparent	134
55	Managing Digital Clock Properties	134
56	Managing RBTI Form Variables	135
57	Managing Table Properties on Demand	136
58	Managing Windows Media Player Properties	137
59	Moving Form Lookup Variables into Custom EEPs	138
60	Persistent Field Value Color in Enhanced DB Grids	141
61	Regular Expressions in DB/Variable Edit Controls	142
62	Searching in the DB Tree View Control	143
63	Selecting All Items in List Box/List View Controls	143
64	Setting the Background Color for Focused Controls	144
65	Storing Comment and Version Information for External Form Files	144
66	Support for Native Form Display	144
67	Suppressing Script Errors in WEB Browser Controls	147
68	Tab Order Sequence in the Form Designer	147
69	Understanding Tab Order	147
70	Understanding the "Space Evenly" Option for Radio Group Controls	148
71	Using a Date Picker (Calendar) in a DB Grid	148
72	Using a Default State for DB/Variable Check Boxes	149
73	Using Bidirectional Mode Properties in R:BASE	149
74	Using Form Default Settings for the Form Wizard	149
75	Using Mask Options with the File List Box Control	150
76	Using Multi-Column Sort in Enhanced DB Grid Control	151
77	Using the "On DB Grid Filter Change" EEP	152
78	Using the FILTERTYPE Property in Enhanced DB Grids	152
79	Using the Enhanced DB Grid Sort Column Properties	152
80	Using the Mouse Over Property for Images	153
81	Using the New Display Format in Forms	153
82	Using the On Increment/On Decrement EEPs for Spinner Controls	154
83	Using the "On Right-Click" EEP on List Box/List View Controls	154
84	Using the SCALEBY Property	155
85	Using the File List Box Control	155
86	Using the Directory Tree Control	156
87	Using the Multi-Select Option for the List View Control	156
88	Using the Object List Toolbar	157
89	Using the "On Key Press" EEP for Instant Results	157
90	Using the Show Accelerator Character Setting	157

91 Using Zebra Stripes in List View Controls	158
Part IX Reports and Labels	160
1 Altering the CrossTab Grid Color	160
2 Capturing Report System Variables	160
3 Capturing the Report Page Number	161
4 Changing the Report Section for All Expressions At Once	161
5 Copy and Paste Expression Builder Variables	161
6 Customizing Column and Row Titles for CrossTab Reports	162
7 Customizing Print Preview Screens with Themes	163
8 Defining Built-in PAUSE Messages in Reports	163
9 Documenting Report/Label EEPs	164
10 Enabling Consolidated Control Palette in Report/Label Designer	165
11 Getting the Total Number of Pages in a Report	165
12 Managing the Memo Control	165
13 Printing a Portion of a Report to a Different Printer	166
14 Printing Watermark Images in RTF Reports	166
15 Retrieving Report Properties	166
16 Sorting Slave Tables with Forms/Sub-Reports	167
17 Using BRUSH_STYLE and PEN_STYLE Properties in Reports	168
18 Using Design & Print Password Options for Reports/Labels	169
19 Using Printer Control Codes	170
20 Using Scientific Notation for Display Format	170
21 Using the Persistent Space Property	171
22 When to Use Report Actions vs. Report Band EEPs	172
Part X File Gateway	174
1 Defining Column Count When Importing Data	174
2 Defining Header/Footer when Using GATEWAY EXPORT XLSW	174
3 Import XLS/XLSX Spreadsheet Column Mapping	175
4 Multi-Tab Excel Workbook with Headers Using GATEWAY EXPORT XLSW	175
5 Suppressing GATEWAY IMPORT Messages	177
6 Using File Gateway Specification Files	177
Part XI R:BASE Editor	180
1 Commenting and Uncommenting Commands	180
2 Customizing REdit Structure Toolbar Options	181
3 Integrating the External R:BASE Editor	184
4 New File Encoding Options	184
5 Running Your Code Within REdit	184

6	Wrapping Commands Using the Right Margin	185
Part XII Trace Debugger		187
1	Accessing Database Settings While in TRACE Mode	187
2	Clearing Error List	187
3	Managing Watch Variables in TRACE Mode	187
Part XIII BLOB Editor		189
1	Customizing the BLOB Editor	189
2	Defining Default Settings for R:BASE BLOB Editor	189
Part XIV Commands		192
1	Assigning Line Numbers using the SELECT Command	192
2	Attaching Server Tables as TEMPORARY Tables	192
3	Automatic Resizing for Custom Images	193
4	Capturing Screenshots to the Windows Clipboard	193
5	Changing CASCADE Options	194
6	Changing Report Margins Using the PRINT Command	194
7	Clearing Stubborn Table Locks	195
8	Creating/Editing RTF Files Outside the Database	195
9	Creating Secure and Encrypted Files	196
10	Customizing Buttons for PAUSE and DIALOG Commands	197
11	Customizing DIALOG POPUP Colors	198
12	Customizing the PRNSETUP Dialog	200
13	Defining Font Name/Size with OUTPUT PRINTER	201
14	Differentiating Between Regular and Temporary Tables/Views	202
15	Displaying Longer Table/Column Names	203
16	Dropping Multiple Views	203
17	Encrypting an UNLOADED DATA File	203
18	Enhanced DIALOG Command	204
19	Enhanced LIST Command Parameters	206
20	Enhanced LIST FORMS, REPORTS and LABELS Command	206
21	JOIN, INTERSECT, SUBTRACT, and UNION - Temporary Table Support	207
22	LAUNCHing Operations with Show Modes	207
23	Listing Database Keys	208
24	Listing Tables with Cascade and Cascade Type	208
25	Managing Monitors and Monitor Properties	208
26	Programmatically Clean Scratch Files on Exit	209
27	RECALCulating Specific Variables	209
28	RUN filespec vs. QUIT TO filespec	209

29	Saving the Clipboard Contents to an External File	210
30	Selecting Available Printers for R:BASE Sessions	210
31	Selecting the Appropriate Tray Using PRNSETUP	211
32	Turning the Data Browser into a Quick Form	212
33	Understanding Dotted vs. Ampersand Variables	212
34	Understanding WHILE Loop Optimization	215
35	Using Additional Data Browser/Editor Options	215
36	Using an Enhanced Calculator in a Dialog Box	216
37	Using AS ALIAS and CAPTION With External Forms	217
38	Using DROP ALL	218
39	Using INPUT Color for the DIALOG Command	218
40	Using INITPOS and INITVAL OPTIONS in CHOOSE Commands	219
41	Using ON CONNECT	223
42	Using PRNSETUP Help Options	224
43	Using the Default Tab for the R:BASE BLOB Editor	225
44	Using the Enhanced ATTACH Command	226
45	Using the Enhanced DIR Command in R:BASE	226
46	Using the GAUGE Options with PAUSE	227
47	Using the LAUNCH Command	228
48	Using the New RUNRBA Command	228
49	Using the PRNSETUP OPTION STANDARD Parameter	228
50	Using the SCALEBY Property in R:BASE Application (.RBA)	229
51	Using the TOP and LEFT coordinates for PRNSETUP	229
52	Using the Zoom Level Percent for PRINT	230
53	Using Themes for the #WHERE Builder	230
54	Using SET FEEDBACK	230
55	Using SET PROGRESS	231
56	Using Wild Cards with PROPERTY	232
57	Validating An Internet Connection	232

Part XV Functions

234

1	Capturing PRNSETUP INFO Results	234
2	Defining First and Last Date of the Current Month	234
3	Finding PORTS on Your Workstation	235
4	Finding Relative Word Position in a Sentence	235
5	Finding Table/View Locks Programmatically	236
6	Finding the Data Type for a given Column or Variable	236
7	Finding the INTEGER value for a given TEXT String	237
8	Managing Application Security Using New CVAL Functions	237
9	Managing DLCALL Functions	238

10	Rounding Down with RNDDOWN Function	238
11	Understanding the New IFF Logical Function	239
12	Using Additional Logical Functions	241
13	Using Enhanced ICAP2 Function	241
14	Using IFWINDOW	242
15	Using IINFO	242
16	Using CHKCUR	246
17	Using CHKVAR	246
18	Using SLOCP	247
19	Using SLOC1 Function to Locate the Number of String Instances	247
20	Using the Enhanced (GETDATE(' ')) Options	248
21	Using the FORMAT Function with .#NOW	250
22	Using the LISTOF Function for Large Data	251
23	Using TINFO	252

Part XVI R:Compiler 255

1	Customizing the Load Window in Compiled Applications	255
2	Is the Running Application Compiled or Not?	257
3	Managing Embedded Resources for Compiled Applications	257
4	Reloading Resource Files	258
5	Saving/Recycling R:Compiler Projects	259
6	Understanding R:Compiler Resources	259

Part XVII Plugins 262

1	Additional R:Spell Checker Dictionaries on a Network	262
2	Capturing Common Operating System Folders	264
3	Converting RTF to Plain TEXT, NOTE or VARCHAR	265
4	Creating a Playlist for Windows Media Player	267
5	Dynamically Assigning R:Charts' Tables/Views	268
6	Finding the Row Size using the RRowSize Plugin	269
7	Sending E-Mail Through MS Outlook With R:BASE MAPI Mail Plugins	271
8	Using Enhanced Appearance Options In R:Charts	272
9	Using the Enhanced MAPIMail Plugins with Thunderbird	273
10	Using the R:AudioPlayer Plugin	274

Part XVIII Oterro / ODBC 277

1	Checking Version and Build of Oterro	277
2	Configuring Oterro and R:BASE ODBC Drivers on 64-bit OS	277
3	Using UPDATE Triggers with Oterro	279
4	Which ODBC Driver to Use	280

Part XIX Feedback

283

Index

284

Part

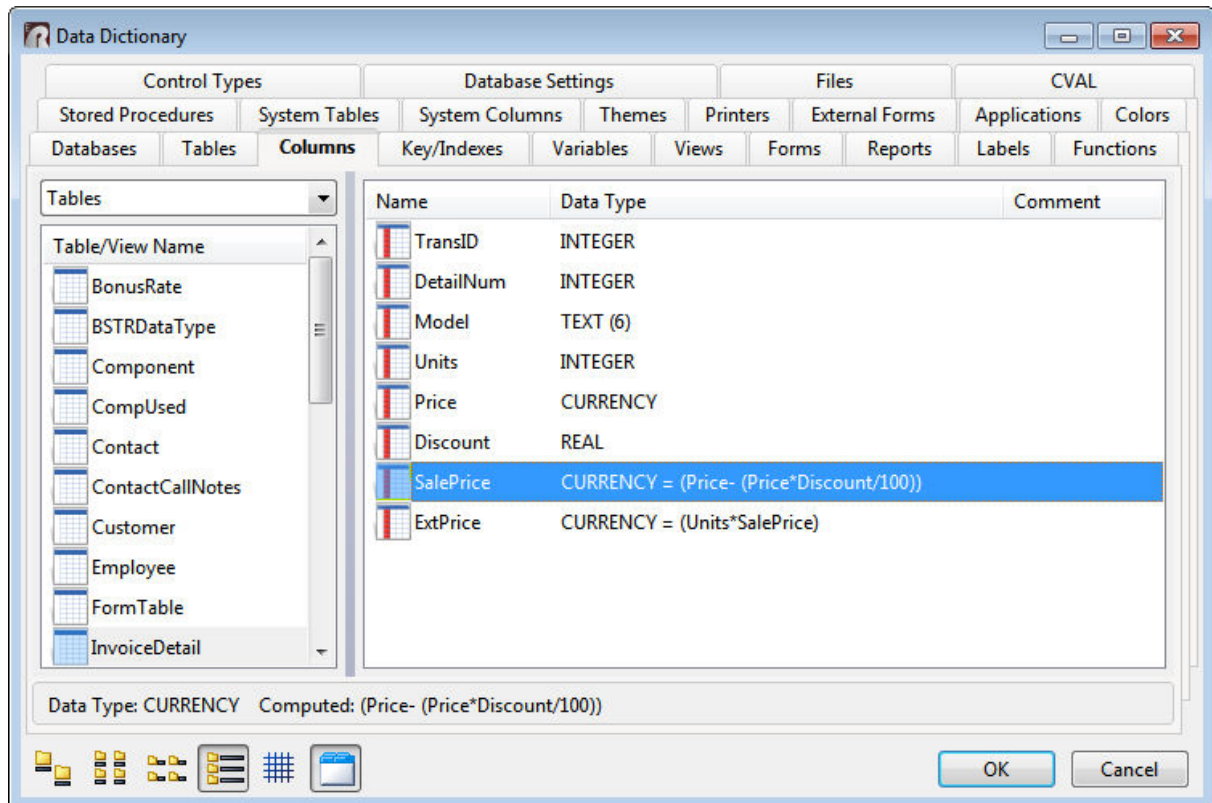


1 General

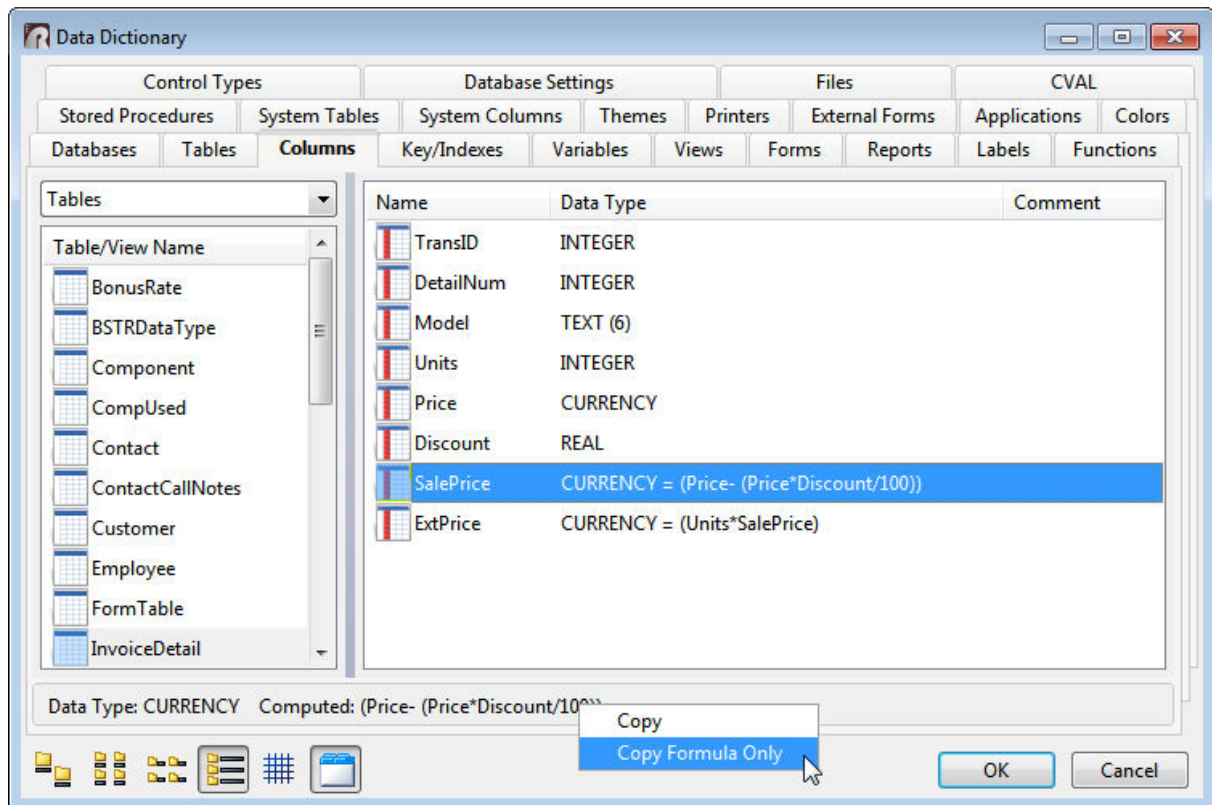
1.1 Capturing Computed Column Expressions

Did you know that when a computed column is displayed in the Data Dictionary, it is possible to copy the entire column definition or just the formula?

While using the R:BASE Editor, pressing the [F3] key, and selecting a specific computed column, the expression/formula is listed.



By then using the right-click mouse button, a menu is available to copy the complete column definition or only the formula.



Selecting either option will place the text in the clipboard.

Please Note:

You will have to click on the [Cancel] button and NOT the [OK] button to retain the copied "formula" to paste into the R:BASE Editor, or any other source.

Pressing the [OK] button will overwrite previously copied string with currently selected column(s), as intended with the Data Dictionary.

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.2 Configuring Anti-Virus Programs to Work with R:BASE

Client and server computers may have one or more anti-virus, anti-malware, and/or anti-spyware utilities installed.

If R:BASE itself is scanned each time it is launched, and the contents of the database and temporary files are halted and scanned each time they are created/accessed, users will see much slower response times when running R:BASE, making connections, and accessing the files.

When using any security-based software, it is important to add exceptions to three areas of the R:BASE program:

R:BASE Program Executable

For R:BASE X (Version 10), the executable name is RBGX.exe. The default folder where it is installed is "C:\RBTI\RBGX".

For R:BASE X Enterprise (Version 10), the executable name is RBGXE.exe. The default folder where it is installed is "C:\RBTI\RBGXE".

For R:BASE eXtreme 9.5 (32), the executable name is RBG95_32.exe. The default folder where it is installed is "C:\RBTI\RBG95_32".

For R:BASE eXtreme 9.5 (64), the executable name is RBG95_64.exe. The default folder where it is installed is "C:\RBTI\RBG95_64".

Database Files

For R:BASE X (Version 10) and eXtreme 9.5 (32), the database files are the four .RB1, .RB2, .RB3, and .RB4 files.

For R:BASE X Enterprise (Version 10) and eXtreme 9.5 (64), the database files are the four .RX1, .RX2, .RX3, and .RX4 files.

Although wildcard may be accepted, adding each file extension individually as an exception may be more appropriate.

Temporary Scratch Files

When R:BASE is launched, it will create three .\$\$\$ temporary files. After performing certain actions, additional files may be created, and the existing files will increase and decrease in size. Add the *.\$\$\$ files as an exception.

1.3 Copying Forms, Labels, and Reports

Did you know that you can copy any number of selected Forms, Labels, and Reports from one R:BASE database to another R:BASE database by simply copying [Ctrl+C] from the source database and pasting [Ctrl+V] to the destination database while both databases are connected as two separate sessions of R:BASE?

You can use the same techniques of copying and pasting between R:BASE X and R:BASE X Enterprise sessions as well!

In addition, you may copy (unload) any number of selected Tables, Views, Stored Procedures, Forms, Labels, Report, External Forms, and Command Files from R:BASE Database Explorer to any folder or even your desktop, by simply dragging and dropping the selected item(s).

Hint:

While copying and pasting, make sure that both sessions of R:BASE are using the similar modules in Database Explorer. For example, copying Forms from one Database Explorer to another Database Explorer you will have to make sure that the Forms section of the destination database is opened before pasting the forms. Thus, Forms to Forms, Labels to Labels, and Reports to Reports, and so on.

1.4 Customized Input Language for the ROSK

The R:BASE On-Screen Keyboard (ROSK) is a visual keyboard displayed on the screen that allows users to type and enter data, instead of relying on a physical keyboard.

With all standard keys displayed, users can select keys using the mouse or another pointing device. The PROPERTY command can be used to customize the R:BASE On-Screen Keyboard button font color, button background color, button edge, background color, screen location, and window caption. The keyboard can also be resized by dragging window sides. When altered, the size and position is preserved between sessions.

The R:BASE On-Screen Keyboard (ROSK) is accessible using the following methods:

01. Selecting "Utilities" > "On Screen Keyboard" from the main Menu Bar
02. Pressing the [Ctrl]+[Alt]+K key combination
03. Using the PROPERTY command

```
PROPERTY ROSK SHOW TRUE/FALSE
```

Did you know that now you can customize the ROSK using an Input Language option?

New "Input Language" setting has been added to alter the keyboard input language within R:BASE. Keyboard types for languages and custom setups are available.

Languages can also be specified using a language name or an 8-digit input locale with the PROPERTY command.

```
-- Example 01: To use a named language for Russian:
PROPERTY ROSK SHOW TRUE
PROPERTY APPLICATION INPUT_LANGUAGE RUSSIAN

-- Example 02: To use the 8-character locale for Albanian:
PROPERTY ROSK SHOW TRUE
PROPERTY APPLICATION INPUT_LANGUAGE 0000041C
```

For additional information on input language, enter "HELP INPUTLANGUAGE" at the R>.

For complete details and more options/examples, refer to the latest R:Docs X.

Caution:

After using the "PROPERTY APPLICATION INPUT_LANGUAGE value" command for testing, make sure to always revert back to your native OS language.

Supported Versions:

- R:BASE X (Build..: 10.0.1.10727 or higher)

1.5 Customizing the R:BASE On-Screen Keyboard (ROSK)

Designing and deploying a custom and colorful R:BASE On-Screen Keyboard (ROSK) is among the cool features when developing R:BASE applications for Mobile/Tablet PCs with different screen resolutions.

Did you know that now you can customize the size (height and width) of the ROSK, based on the Mobile/Tablet PCs display size?

R:BASE includes the enhancement to customize the HEIGHT and WIDTH of the ROSK.

```
-- Example 01:
-- Customizing HEIGHT and WIDTH for ROSK
PROPERTY ROSK HEIGHT 350
PROPERTY ROSK WIDTH 700
PROPERTY ROSK SHOW TRUE
RETURN
```

The keyboard can also be resized by dragging window sides, which will be stored and remembered as ROSK settings when using the "PROPERTY ROSK SHOW TRUE" next time.

A technical document "Using the R:BASE On-Screen Keyboard" is also available at:

From The Edge: <http://www.razzak.com/fte/>

Topic: Using the R:BASE On-Screen Keyboard (ROSK) in R:BASE X (Version 10)

Supported Versions:

- R:BASE X (Build..: 10.0.1.10522 or higher)

1.6 Customizing ROSK for R:BASE Mobile Applications

The R:BASE On-Screen Keyboard (ROSK) is a visual keyboard displayed on the screen that allows users to type and enter data, instead of relying on a physical keyboard. With all standard keys displayed, users can select keys using the mouse or another pointing device. The PROPERTY command can be used to

customize the R:BASE On-Screen Keyboard button font color, button background color, button edge, background color, screen location, and window caption.

The R:BASE On-Screen Keyboard (ROSK) is accessible using the following methods:

1. Hot Keys: Pressing the [Ctrl]+[Alt]+K key combination
2. Database Explorer | Main Menu | Utilities | On-Screen Keyboard
3. Using the PROPERTY command: PROPERTY ROSK SHOW 'TRUE'

Did you know that you can provide a Numeric-Only Keyboard, disable any number of keys, change the caption, and specify a custom hot key to display or hide the keyboard?

You can provide a Numeric-Only Keyboard, disable any number of keys, change the caption, and specify a custom hot key to display or hide the keyboard as follows:

-- Example 01 (To display a Numeric-Only R:BASE On-Screen Keyboard)

```
PROPERTY ROSK SHOW 'TRUE'  
PROPERTY ROSK TYPE NUMPAD
```

-- Example 02 (To display the Alpha-Numeric R:BASE On-Screen Keyboard)

```
PROPERTY ROSK SHOW 'TRUE'  
PROPERTY ROSK TYPE ALPHA
```

For complete details and implementing R:BASE On-Screen Keyboard (ROSK) for your R:BASE mobile applications running on Tablet PCs and/or Touch-Screens, refer to:

From The Edge: <http://www.razzak.com/fte/>

Topic: Using the R:BASE On-Screen Keyboard (ROSK) in R:BASE X (Version 10)

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.7 Data Dictionary and Selecting Color Value

Data Dictionary [F3] Key in R:BASE provides the following valuable information at your fingertips:

- Databases
- Tables *
- Columns *
- Key/Indexes *
- Variables
- Views *
- Forms *
- Reports *
- Labels *
- Functions
- Stored Procedures *
- System Tables *
- System Columns *
- Form Component IDs **
- Form Actions **
- Report Component IDs ***
- Label Component IDs ****
- Themes
- Printers
- External Forms
- Applications
- Colors
- Control Types
- Database Settings
- Files

* = Connected Database

** = While in Form Designer

*** = While in Report Designer
 **** = While in Label Designer

These are very slick and handy options that allow users to capture information about the R:BASE environment and/or connected database.

These options can be accessed from anywhere in R:BASE by pressing the [F3] key or selecting "Data Dictionary" option from the Menu Bar under "Tools".

While programming and customizing applications, if you wish to use custom font color, background color, for CHOOSE, DIALOG, or PAUSE commands you can press the [F3] key, select the color you like and then either double-click for single item or [OK] button for multi-select items. Using this approach, you can capture the name of the color listed in the first column.

The Data Dictionary | Color tab includes a list of 190 colors.

Did you know that now you can choose which option you want as the return value?

Here's how:

Press [F3] Data Dictionary Key and notice the additional option for [Return Value].

You now have four (4) options to pick from.

[] Color [] Name [] Integer Value [] RGB Value

Supported Versions:

- R:BASE X (Build..: 10.0.1.10522 or higher)

1.8 Defining a Default Path for "Open Image" Dialogs

Did you know that you can define a folder path where all "Open Image" dialogs will initially start?

A new RBTI System Variable, RBTI_IMG_FOLDER, is now supported to define the valid folder path where all "Open Image" dialogs will launch.

```
SET VAR RBTI_IMG_FOLDER TEXT = 'C:\TEMP\IMAGES\'
```

Notes:

- The RBTI_IMG_FOLDER System Variable must be redefined for a new session of R:BASE, so define the variable in your development startup file.
- If the RBTI_IMG_FOLDER variable does not exist or contains a non-valid folder path, all image dialogs will open in the current folder.

Supported Versions:

- R:BASE X (Build..: 10.0.1.10522 or higher)

1.9 Deploying an R:BASE X Enterprise Database

When you are ready to deploy your first R:BASE X Enterprise database and application, you may want to reset the version and time stamp of all your forms, labels, reports, and stored procedures to clean start the first deployed version of your application.

Use the following code:

```
-- TimeStamp_Forms_Labels_Reports_StoredProcedures.RMD
```

```
UPDATE SYS_FORMS3 SET SYS_FORM_VERSION = 1, SYS_FORM_MOD_TS = .#NOW
UPDATE SYS_LABELS3 SET SYS_LABEL_VERSION = 1, SYS_LABEL_MOD_TS = .#NOW
UPDATE SYS_REPORTS3 SET SYS_REPORT_VERSION = 1, SYS_REPORT_MOD_TS = .#NOW
UPDATE SYS_PROCEDURES SET SYS_PROC_VERSION = 1, SYS_PROC_MOD_TS = .#NOW
```

RETURN

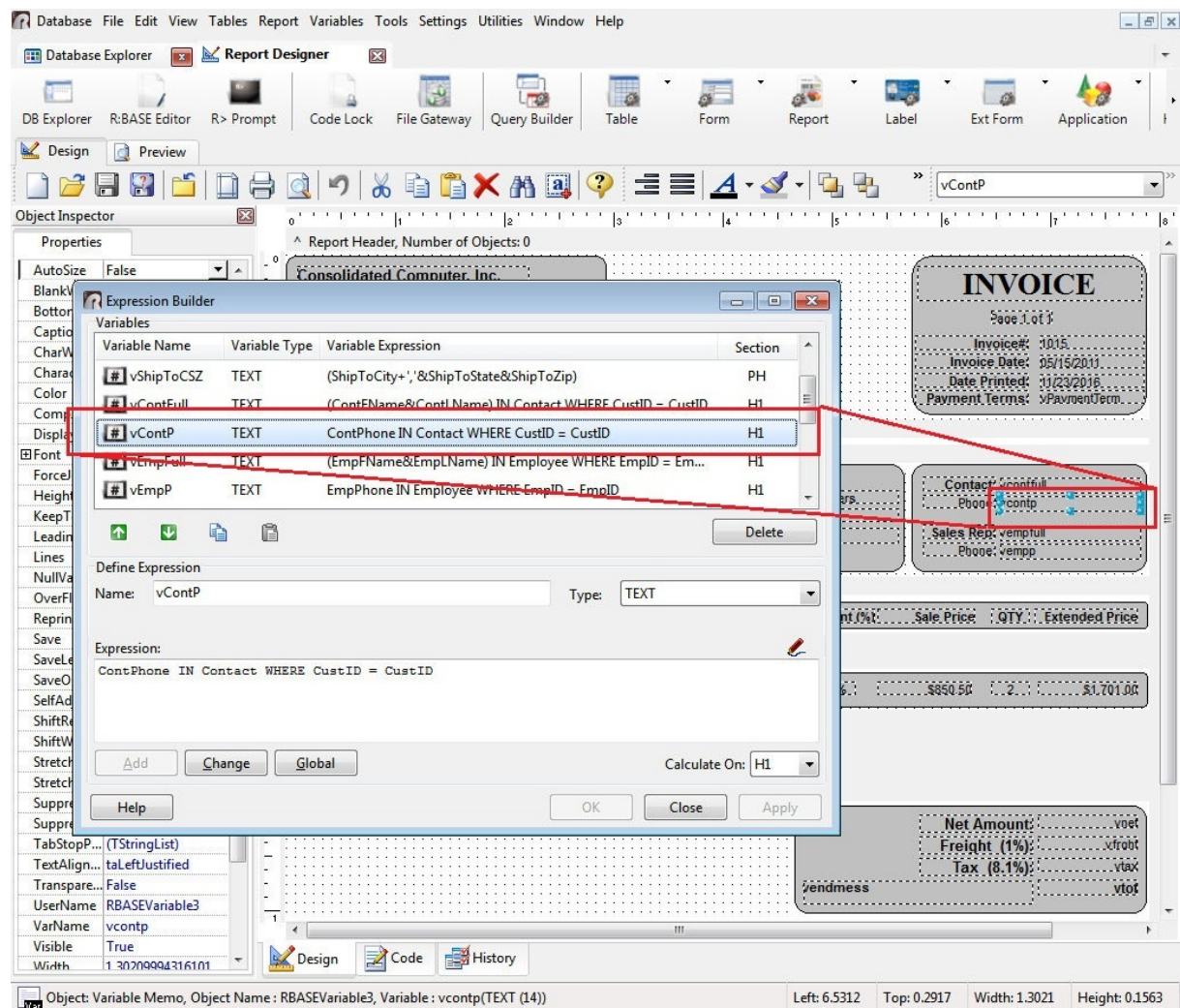
Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.10 Display the Selected Expression Builder Variable

Did you know that you can now see the highlighted variable object within the Form/Report/Label Designer canvas background, when editing variables within the Expression Builder?

This feature allows for easier recognition of a variable's location while reviewing the expression definition. Switching to different variables in the Expression Builder list will move the blue handlebars to that variable object.



Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.11 Displaying R:BASE Keys/Indexes

While in Data Designer (RBDefine), the [Key/Indexes] option allows users to create table constraints; such as Primary Keys, Foreign Keys and Unique Keys. Users can also create Indexes, which provide faster data retrieval for the database.

Did you know that you can retrieve and display a complete list of Keys and Indexes, if defined, using the Data Dictionary (F3) key?

Among many other database and non-database information, the enhanced Data Dictionary (F3) key in R:BASE, also includes a tab for [Keys/Indexes]. Selecting [Keys/Indexes] tab will list all database Keys and Indexes, if defined.

By default, all Keys and Indexes are displayed including details such as, ID, Type, Table Name, and References. Clicking on the appropriate column header will sort the column in ascending and descending order accordingly.

However, if you wish to only display Primary Keys, Unique Keys, Indexes, or Foreign Keys, you may use the Type: Drop-Down Filter option located above the columns to specifically display your selected option of Primary Keys, Unique Keys, Indexes, or Foreign Keys, if you wish.

The Data Dictionary [F3] key includes the following options:

- Databases
- Tables
- Columns
- Keys/Indexes
- Variables
- Views
- Forms
- Reports
- Labels
- Functions
- Stored Procedures
- System Tables
- System Columns
- Themes
- Printers
- External Forms.
- Applications
- Colors
- Control Types
- Database Settings
- Files

While in Form Designer, the Data Dictionary includes the following additional tabs:

- Form Component IDs
- Form Actions

While in Report Designer, the Data Dictionary includes the following additional tab:

- Report Component IDs

While in Label Designer, the Data Dictionary includes the following additional tab:

- Label Component IDs

While in External Form Designer, the Data Dictionary includes the following additional tabs:

- External Form Component IDs
- External Form Actions

1.12 Documenting Component IDs in Forms/Labels/Reports

Did you know that you can document all embedded Custom EEPs in Forms, Labels and Reports using the "Document Custom EEPs" option while in Form, Label or Report Designer?

Form Designer > main Menu Bar > Form > Document Custom EEPs
 Label Designer > main Menu Bar > Label > Document Custom EEPs
 Report Designer > main Menu Bar > Report > Document Custom EEPs

Notice the following output options:

- Printer
- Text File
- PDF File
- Clipboard
- **In Text Editor**
- **EEP Map**

The complete documentation of embedded Custom EEPs will also include the Component IDs, if defined.

Supported Versions:

- R:BASE X (Build..: 10.0.1.10522 or higher)

1.13 Dynamically Changing Themes in R:BASE Application (.RBA)

Current versions of R:BASE for Windows allow you to design and distribute the stand alone R:BASE Application (.RBA) files using the "Applications" module of R:BASE Database Explorer.

While in the Application designer, you may elect to use an Application Theme by checking the [] Use Theme option and then selecting your favorite theme from among the list of supported themes. Did you know that now you can dynamically change the R:BASE Application (.RBA) theme using the following PROPERTY command?

Example 01:

```
-- Start
-- Application: On After Start Action
CLEAR VARIABLE vThemeName
SET VAR vThemeName TEXT = NULL
IF (CVAL('NetUser')) <> 'Razzak' THEN
  PROPERTY RBA_FORM THEMENAME 'Sports Blue'
  PROPERTY RBA_FORM CAPTION 'Using Sports Blue'
ELSE
  PROPERTY RBA_FORM THEMENAME 'R:BASE Rocks!'
  PROPERTY RBA_FORM CAPTION 'Using R:BASE Rocks!'
ENDIF
RETURN
-- End
```

Example 02:

```
-- Start
-- Application: On After Start Action
IF (CVAL('DATABASE')) <> 'RRBYW19' OR (CVAL('DATABASE')) IS NULL THEN
  CONNECT RRBYW19 IDENTIFIED BY NONE
ENDIF
CLEAR VARIABLE vUserThemeName, vThemeCaption, vDefaultTheme
SET VAR vUserThemeName TEXT = NULL
SET VAR vThemeCaption TEXT = NULL
SET VAR vDefaultTheme TEXT = 'Convergence'
SELECT UserThemeName, ThemeCaption INTO +
  vUserThemeName INDIC ivUserThemeName, +
  vThemeCaption INDIC ivThemeCaption +
  FROM UserThemes WHERE UserName = (CVAL('NetUser'))
```

```

IF vUserThemeName IS NULL THEN
  SET VAR vUserThemeName = .vDefaultTheme
  SET VAR vThemeCaption = .vDefaultTheme
ENDIF
PROPERTY RBA_FORM THEMENAME .vUserThemeName
PROPERTY RBA_FORM CAPTION .vUserThemeName
RETURN
-- End

```

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.14 Dynamically Defining Unique File Names

Here are two examples to create a unique file name in a multi-user and multi-session environments using R:BASE.

```

-- Example 01
-- Using R:BASE X (Version 10)
IF (CVAL('DATABASE')) <> 'RRBYW19' OR (CVAL('DATABASE')) IS NULL THEN
  CONNECT RRBYW19 IDENTIFIED BY NONE
ENDIF
-- Verify the existence of PDF Sub-Directory
SET VAR vChkFile INTEGER = NULL
SET VAR vChkFile = (CHKFILE('PDF'))
IF vChkFile <> 1 THEN
  MD PDF
ENDIF
CLS
-- Get Globally Unique ID
SET VAR vGUID TEXT = NULL
PLUGIN RGUID vGUID|SYMBOLS OFF
-- Create Unique PDF File Name
SET VAR vPDFFileName TEXT = NULL
SET VAR vPDFFileName = +
('PDF\'+(CVAL('NetUser'))+'_'+vGUID+'_CustomerList_' +
(FORMAT(.#DATE,'MMDDYYYY'))+'.pdf')
PRINT CustomerList +
OPTION PDF +
|FILENAME .vPDFFileName +
|SHOW_CANCEL_DIALOG ON +
|TITLE Customer List +
|SUBJECT Customer List +
|AUTHOR R:BASE X +
|KEYWORDS Customer List +
|OPEN ON
CLEAR VARIABLES vChkFile,vGUID,vPDFFileName,RBTI_%
RETURN

-- Example 02
-- Using R:BASE X Enterprise (Version 10)
IF (CVAL('DATABASE')) <> 'RRBYW19' OR (CVAL('DATABASE')) IS NULL THEN
  CONNECT RRBYW19 IDENTIFIED BY NONE
ENDIF

```



```
-- Verify the existence of PDF Sub-Directory
SET VAR vChkFile INTEGER = NULL
SET VAR vChkFile = (CHKFILE('PDF'))
IF vChkFile <> 1 THEN
MD PDF
ENDIF
CLS
-- Create Unique PDF File Name
SET VAR vPDFFileName TEXT = NULL
SET VAR vPDFFileName = +
('PDF\'+(CVAL('NetUser'))+\'_'+(CVAL('GUID'))+\'_CustomerList_\'+ +
(FORMAT(.#DATE, 'MMDDYYYY'))+\'_\.pdf')
PRINT CustomerList +
OPTION PDF +
|FILENAME .vPDFFileName +
|SHOW_CANCEL_DIALOG ON +
|TITLE Customer List +
|SUBJECT Customer List +
|AUTHOR R:BASE X Enterprise +
|KEYWORDS Customer List +
|OPEN ON
CLEAR VARIABLES vChkFile,vPDFFileName,RBTI_%
RETURN
```

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.15 Enabling Object Inspector

Did you know that R:BASE includes the built-in Object Inspector for Form, Label and Report Designers?

While in Form, Label or Report Designer, you can enable the Object Inspector to explore in-depth properties of Form, Label, Report, and any selected object.

Here's how to enable the Object Inspectors in R:BASE eXtreme:

Form Designer > Menu Bar > View > Toolbars > Object Inspector

Report Designer > Menu Bar > View > Toolbars > Object Inspector

Label Designer > Menu Bar > View > Toolbars > Object Inspector

Once enabled, you may dock the Object Inspector on the left, or float anywhere you find more convenient!

1.16 External Print Module Within the TabPrint.dll

Did you know the R:BASE Data Browser has been enhanced with a WYSIWYG format print module for data output?

In the recent public updates, R:BASE Technologies removed this print module from the R:BASE program and is now as a separate DLL file, TabPrint.dll. The original R:BASE setup had the DLL embedded in the RBASE executable, and extracted on demand.

With intentions in making the R:BASE executable smaller and to support a faster startup, the TabPrint.dll is now stored in the R:BASE/Runtime/R:Compiler program folder. Also, as the DLL is "extracted on demand" when the print option is selected, it is preferred to be stored as an external file to avoid causing

false positives for security software.

For R:Compiler and Runtime compiled applications, please include the TabPrint.dll to support the WYSIWYG print module within the Data Browser. Copy the TabPrint.dll from your R:Compiler/Runtime program directory to the custom program directory.

Supported Versions:

- R:BASE X (Build..: 10.0.1.10627 or higher)

1.17 Finding the Exact Name of a Form Control Type

When creating the Dynamic Objects for Forms, using the new CREATEOBJECT command, you have to know the exact name for the Control Type to be created dynamically at runtime.

Syntax:

```
CREATEOBJECT +  
<ObjectType> <ComponentID> <ParentControl> <Left Top Width Height>
```

While in the Form Designer, *every* property of the selected object including the "Name" and "Object Type" is listed under the Object Inspector.

Did you know that you can use the new [Control Types] tab of the Data Dictionary [F3] to highlight and paste the exact name for the desired Object Type?

In addition to specific tabs that become visible within which module the [F3] key was pressed, you will always find the [Control Types] tab from any module of R:BASE.

Data Dictionary [F3] General Tabs:

- System Tables
- System Columns
- Themes
- Control Types
- Tables
- Columns
- Variables
- Views
- Forms
- Reports
- Labels
- Functions
- Stored Procedures

Data Dictionary [F3] Form Designer Specific Tabs:

- All General tabs listed above, plus ...
- Form Component IDs
- Form Actions

Data Dictionary [F3] Label Designer Specific Tabs:

- All General tabs listed above, plus ...
- Label Component IDs

Data Dictionary [F3] Report Designer Specific Tabs:

- All General tabs listed above, plus ...
- Report Component IDs

Supported Versions:

- R:BASE X (Build..: 10.0.1.10522 or higher)

1.18 Handling Milliseconds in Parameterized Update Statements

When converting MS SQL or Oracle databases to R:BASE X Enterprise, or using parametrized update statements, it takes the hours, minutes, and seconds, but drops off the milliseconds.

By default, the milliseconds are not set in the configuration file of R:BASE and Oterro. However, if you wish to deal with milliseconds, you will need to add that option in R:BASE and Oterro configuration files accordingly.

Here's how:

In Oterro configuration file, change the TIME FORMAT (line 48) to HH:MM:SS.SSS and TIME SEQUENCE (line 49) to HHMMSSSSS. Additional ".SSS" are used to handle milliseconds.

Example:

```
TIME FORMAT HH:MM:SS.SSS
TIME SEQUENCE HHMMSSSSS
```

Based on the R:BASE version, follow the same routine for appropriate R:BASE configuration file(s).

In addition, make sure that the connected R:BASE database settings are also configured accordingly.

1.19 How Healthy is your Database?

Have you ever wondered how healthy your database really is? Are all defined indexes properly indexed? Are there any duplicate indexes? Are all Primary/Foreign/Referenced keys valid? Are all defined default column values still valid? Are all defined views still functional? Is there any inconsistency in performance?

These issues become important concerns when upgrading/migrating from an old (LEGACY) DOS or Windows version of R:BASE. Technically, there is very little to worry about when your database is designed and managed using the latest Data Designers (RBDefine), Enhanced CREATE and LOAD commands of R:BASE that fully enforce all rules and validate everything before the table is created/saved or the data is loaded. Granted, all things must be considered.

However, if you have upgraded your legacy database and everything seems to be working fine, do yourself and your lovely clients a big favor! Perform one additional task to validate the integrity of your database.

Follow the steps below and see if everything is "peachy".

1. Make a backup copy of the database.
2. Copy the entire database (.RB1-.RB4/.RX1-.RX4 files) on your local drive in a separate folder with at least twice as much available free disk space.
3. Start R:BASE using the latest version/update and switch the current folder to the appropriate database folder on your local drive.
4. At the R> prompt:

```
DISCONNECT
SET MESSAGES ON
SET ERROR MESSAGES ON
SET MULTI OFF
-- CONNECT the database with OWNER password, if any
CONNECT dbname IDENTIFIED BY owner password
-- CONNECT the database with no passwords
CONNECT dbname
```

5. Check the connected database CHARacter settings, especially the IDQUOTES. At the R> prompt:

```
CLS
```

```
SHOW CHAR
```

Notice the setting for IDQUOTES (the last item on list). If this setting is blank/null, make sure to set the IDQUOTES settings to ` (that is a single reversed quote, on the same key as the ~ tilde).

```
SET IDQUOTES= `
```

6. Now create the unload file with NULL set to -0-. At the R> prompt:

```
CLS
SET NULL -0-
OUTPUT newdb.ALL
UNLOAD ALL
OUTPUT SCREEN
```

This step will create two files (newdb.ALL and newdb.LOB)

7. DISCONNECT, and then rename this database to some other name. At the R> prompt:

```
DISCONNECT
RENAME dbname.RB? dbnameBK.RB?
```

8. Now, let's rebuild the fresh database and see if it passes the integrity check. At the R> prompt:

```
CLS
RUN newdb.ALL
```

Watch the activity and all messages on the screen. Don't fall asleep! You might miss an important warning. Completion time may vary based on size of the database (number of tables/records/indexes).

If there were no warnings or error messages, you've got a fresh/healthy database. Give yourself a pat on the back.

If there are any warning or -ERROR- message(s), take them seriously. In this case, DISCONNECT the bad database (built using RUN newdb.ALL). At the R> prompt:

```
DISCONNECT
```

Delete the bad database. At the R> prompt:

```
DELETE dbname.rb?
DELETE newdb.ALL
DELETE newdb.LOB
-- Rename the previously saved database (Step 07.)
RENAME dbnameBK.RB? dbname.RB?
-- CONNECT to the database
CONNECT dbname
```

9. CORRECT all -ERROR-s accordingly and then repeat Step 06.
10. Do not quit or give up until you see a completely fresh database without any warnings or errors.

Believe me, the time spent on this exercise will save you a lot of hassle and embarrassment, instead of chasing something obvious that should have been taken care of a long time ago. Hope that helps!

Additional housekeeping and database maintenance tips are also available on From The Edge:
<http://www.razzak.com/fte/>

1.20 Implementing Application Privileges

Traditionally, for user application access, you could enable or disable almost any form control by using the (CVAL('NetUser')), (CVAL('Computer')), (GETVAL('GetIPAddress','n')), (GETVAL('GetMACAddress','n')), and so on ..

However, it would be very time consuming as well as redundant if you were to provide application access based on the individual user. You could still implement that logic if only a few items in your menu structure are accessible to a few users.

Did you know that now you can simply use the new (CVAL('NETLOCALGROUP')) and (CVAL('NETGROUP')) functions in R:BASE eXtreme 9.5 to accomplish the goal of providing your application privileges at the "group" level?

A new (CVAL('NETGROUP')) function retrieves a list of global groups to which a specified user belongs.

A new (CVAL('NETLOCALGROUP')) function retrieves a list of local groups to which a specified user belongs.

If needed, using these new CVAL functions, now you can implement the application privileges based on local or global network groups.

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.21 Is the Application Compiled or Not?

When designing and distributing R:Compiler applications, your main menu options may include a line item to access Database Explorer or the R> Prompt.

Accessing that particular option would be fine if the end user is using the full version of R:BASE. However, that option would go nowhere if the end user is using the compiled version of the custom application.

Did you know that you can assign a unique Component ID to a particular item or button on your application main menu and then use the following GETPROPERTY and PROPERTY command to manage access to that option?

```
-- Start
-- On After Start EEP
-- Know if the running application is compiled or not?
CLEAR VARIABLE vCompiled
SET VAR vCompiled TEXT = NULL
GETPROPERTY APPLICATION ISCOMPILED vCompiled
IF vCompiled = 'YES' THEN
    PROPERTY AccessDatabaseExplorer VISIBLE 'FALSE'
ENDIF
CLEAR VARIABLE vCompiled
RETURN
-- End
```

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.22 Is the Application Runtime or Not?

When designing and distributing Runtime applications, your main menu options may include a line item to access Database Explorer or the R> Prompt.

Accessing that particular option would be fine if the end user is using the full version of R:BASE.

However, that option would go nowhere if the end user is using the Runtime version of the application.

Did you know that you can assign a unique Component ID to a particular item or button on your application main menu and then use the following GETPROPERTY and PROPERTY command to manage access to that option?

```
-- Start
-- On After Start EEP
-- Know if the running application is runtime or not?
CLEAR VARIABLE vRuntime
SET VAR vRuntime TEXT = NULL
GETPROPERTY APPLICATION ISRUNTIME vRuntime
IF vRuntime = 'YES' THEN
    PROPERTY AccessDatabaseExplorerer VISIBLE 'FALSE'
ENDIF
CLEAR VARIABLE vRuntime
RETURN
-- End
```

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.23 Managing Custom Colors

While in Form, Label or Report Designer, defining a background color, when the ordinary color palate does not give the color you need, you always have the choice to opt for "Custom ..." colors. Selecting this option provides you with a selection of 48 Basic Colors as well as the option to define or select from 16 "Custom Colors". Once custom colors are defined and added to the list, only the logged in user on that workstation can access those Custom Colors while in Form, Label or Report Designer.

In an environment of R:BASE Developers and/or Team of R:BASE Contractors and Sub-Contractors, each developer must define those custom colors manually for R:BASE for Windows, in order to follow the application standards. In addition, if you define all those custom colors and for some reason you need to replace a hard drive or re-install R:BASE, you also have to re-define those custom colors when designing Forms, Labels or Reports.

Did you know that you can export your custom defined colors to a text file and then import them back into R:BASE when you need them again, or distribute to your own developers or contractors for consistency purposes?

Here' how:

1. Start R:BASE for Windows
2. Open any Form, Label or Report in designer
3. Select any control and use the Object Property to change background color
4. Select the "Custom ..." from the bottom of the list
5. Define your custom color or Add to existing Custom Colors
6. Click on [OK] button to save Custom Colors
7. Now if you wish to export all defined Custom Colors:

```
Database Explorer | Settings | Custom Colors
Notice the following options:
Save to file ...
Load from File ...
```

8. You may also import from a file that was created in Step 7

You can also import your 16 custom colors defined in previous R:BASE for Windows versions into R:BASE eXtreme without re-defining them again.

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.24 Managing External Themes - R:BASE Themes Unleashed!

Did you know that you can use your own custom-designed or ".msstyles" compatible themes with R:BASE?

R:BASE includes support for external themes, which must have the ".msstyles" file extension. These themes can be acquired from any online resource or from a custom theme you created yourself. After a theme is loaded into the R:BASE session, the theme name will be listed in every available location where themes are specified and will also appear in the Data Dictionary list.

When using external themes, the theme file must be loaded into R:BASE each time the session is launched. So, it can be loaded within the environment set up in a startup file.

In order to load a theme into R:BASE, the PROPERTY command must be used. The following PROPERTY command parameters are to manage external themes in R:BASE:

--Load Theme Example:

```
PROPERTY LOAD_THEME 'NewLuna' LunaBlue.msstyles
```

After this command is issued, the theme "NewLuna" will be available in all locations where a theme can be specified.

--Release Theme Example:

```
PROPERTY RELEASE_THEME NewLuna TRUE
```

After this command is issued, the theme "NewLuna" will be released.

--Change Theme Example:

```
PROPERTY CHANGE_THEME NewLuna LunaXP.msstyles
```

After this command is issued, the theme NewLuna will use the styles defined in the LunaXP.msstyles file. This command parameter is provided to alter the theme file, and to avoid changing the specified "theme name" in every location within the code.

Have fun integrating your own themes in R:BASE!

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.25 Managing Multipage Images

Did you know that you can use the built-in Note/BLOB Field Viewer/Editor to manage multipage VARBIT images, such as DCX, TIFF and GIF files?

The enhanced Note/BLOB Field Viewer/Editor now includes an additional option to manage Multipage Images. When Editing or Browsing the data using Data Browser/Editor, or a DB Grid in forms, the dynamic Note/BLOB Field Viewer/Editor is used if you double click on the NOTE, VARCHAR or VARBIT columns.

When using the form as BROWSE USING formname, EDIT USING formname, or ENTER USING formname, the Note/BLOB Field Viewer/Editor is used when double clicking on DB Memo controls for NOTE data types, the DB Rich Edit controls for VARCHAR data types (RTF data), or DB Image controls for VARBIT data types (images).

Depending on the form control used, tab appropriate window tab of the Note/BLOB Field Viewer/Editor is opened to manage the data accordingly. If you wish to manage Multipage images, such as DCX, TIFF or GIFs, you may select the "Multipage Images" tab of the Note/BLOB Field Viewer/Editor.

You may also use the enhanced RBBEDIT (RBBE) command to manage external files.

```
-- Example 01 (To Manage External Plain Text Data)
RBBE filename.txt
```

```
-- Example 02 (To Manage External RTF Data)
RBBE filename.rtf

-- Example 03 (To Manage Bitmap Files)
RBBE filename.bmp

-- Example 04 (To Manage Multipage TIFFs)
RBBE filename.TIF

-- Example 05 (To Manage Multipage DCX Files)
RBBE filename.cdx

-- Example 06 (To Manage Multipage GIFs)
RBBE filename.gif
```

Supported Versions:

- R:BASE X (Build..: 10.0.1.10522 or higher)

1.26 New Data Types in R:BASE X Enterprise

Several new data types are available within R:BASE X Enterprise, for use when designing tables and when defining variables.

In addition to other R:BASE X Enterprise (Version 10) capabilities, this will help those who are also using R:BASE as a front-end to retrieve and manage data when SCONNECTing/SATTACHing MS SQL, Oracle, and IBM DB2 database tables into R:BASE X Enterprise (Version 10).

The BIGINT data type provides a much larger value, when the existing INTEGER limit has been reached.

Both BOOLEAN and SMALLINT are data types that can be implemented to replace a TEXT (T/F)-type columns or small value INTEGER columns respectively, to help decrease the size of large databases. In short, the data types allows developers to hone the database structure to use the best available data type for the data being stored.

The WIDENOTE and WIDETEXT data types columns are variations of NOTE and TEXT, that support the true storage of Unicode data.

BIGINT

- Holds a 64-bit integer value
- Offers a range of $\pm 999,999,999,999,999,999$
- Delimiters (such as commas) cannot be used in entry
- No length is needed

BOOLEAN

- Hold true/false values
- Internally stored as 0 for false and 1 for true
- Accepted values for false include: 0, false, 'false', f
- Accepted values for true include: 1, true, 'true', t
- For multilingual applications, the values of 0 and 1 are recommended

SMALLINT

- Holds a 16-bit integer value
- Offers a range of ± 32767
- Delimiters (such as commas) cannot be used in entry
- No length is needed

WIDENOTE

- Holds Unicode data
- The default length is 0, where the length is determined by the data
- Holds variable length text of up to 4,092 characters

- Maximum length of a WIDENOTE column can be set
- Indexes and constraints are allowed on WIDENOTE data types
- R:BASE reserves a minimum of four bytes of internal storage space, with 2 bytes per character

WIDETEXT

- Holds Unicode data
- The default length is eight characters
- The maximum is 1,500 characters
- Maximum length of a WIDETEXT column can be set
- R:BASE reserves a minimum of four bytes of internal storage space, with 2 bytes per character

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.27 Optimizing the R:BASE Multi-User Environment

The ability to fine tune R:BASE itself is one of the many enhancements added over the years. Most often, R:BASE is able to make the right decisions about the best way to process commands.

Generally, you know your database and application code better than anyone else. However, you should always take advantage of the latest enhancements and syntax, and would pick a more efficient method if given the opportunity.

Did you know that there are three settings that can improve performance in a multi-user environment?

- STATICDB
- FASTLOCK
- QUALCOLS

STATICDB limits structural changes to the database, and activates a read-only schema mode. When STATICDB is set to ON, permanent changes cannot be made to the database structure (table and column names, for example). New tables can be created, but they are temporary and visible only to the user who created them. By restricting changes to the database structure, R:BASE does not continually re-read the structure to check for changes. Hence, multi-user applications run faster. All users connected to a multi-user database must have the same setting for STATICDB. The default is OFF.

In a multi-user environment, a user who first connects to a database with STATICDB set to ON engages that database to operate in a read-only schema mode, whereby any user must have their STATICDB setting ON in order to connect to that database.

SET FASTLOCK ON for faster multi-user performance while modifying data. With FASTLOCK ON, R:BASE does not place a table lock on the table, allowing for greater throughput. A table lock is only needed to prevent structure changes. FASTLOCK can only be set to ON when STATICDB is set to ON, and both FASTLOCK and STATICDB must be set ON before the database is connected.

QUALCOLS setting is used to turn page locking OFF and use row locking instead and the ROWLOCKS setting is no longer used to turn row locking ON or OFF. If you know that your application mainly updates or deletes data a row at a time, rather than many rows, SET QUALCOLS 2 for row locking.

Use 10 for page locking or row locking as appropriate. Use 2 for row locking only, not page locking. In this case, R:BASE locks a row, reads the row, makes the change, and then releases the row.

Keep in mind that the QUALCOLS setting can be changed dynamically and can be different for different users using the same database. In most cases the SET QUALCOLS 2 is a good combination. However, use SET QUALCOLS 10 for page locking when you are doing an UPDATE and/or DELETE affecting many rows in a table.

Here is a routine that you can use to automate the process of CONNECTing the database in a network environment with SET STATICDB ON, SET FASTLOCK ON, and SET QUALCOLS 2.

```
-- Example
-- Start here ..
-- MyApp.DAT Startup Application File
```

```
-- Start Fresh
CLEAR ALL VARIABLES
LABEL StartFresh
DISCONNECT
SET QUOTES=NULL
SET QUOTES='
SET DELIMIT=NULL
SET DELIMIT=', '
SET LINEEND=NULL
SET LINEEND='^'
SET SEMI=NULL
SET SEMI=';'
SET PLUS=NULL
SET PLUS='+'
SET SINGLE=NULL
SET SINGLE='_'
SET MANY=NULL
SET MANY='% '
SET IDQUOTES=NULL
SET IDQUOTES='`'
SET CURRENCY '$' PREF 2 B
DISCONNECT
SET STATICDB OFF
SET ROWLOCKS ON
SET FASTLOCK OFF
SET TIMEOUT 120
SET FEEDBACK OFF
LABEL Start
DISCONNECT
SET STATICDB ON
SET FASTLOCK ON
SET QUALCOLS 2
SET MESSAGES OFF
SET ERROR MESSAGES OFF
SET ERROR MESSAGE 2495 OFF
CONNECT dbname IDENTIFIED BY ownername
SET ERROR MESSAGE 2495 ON
SET MESSAGES ON
SET ERROR MESSAGES ON
-- Check the availability of database
IF SQLCODE = -7 THEN
CLS
PAUSE 2 USING 'Unable to Connect the Database.' +
CAPTION ' Your Application Caption Here ...' +
ICON WARNING +
BUTTON 'Press any key to continue ...' +
OPTION MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_COLOR RED +
|MESSAGE_FONT_SIZE 11
CLOSEWINDOW
EXIT
ENDIF
-- Enforce Database Default Settings
```

```
SET QUOTES='
SET DELIMIT=' , '
SET LINEEND='^'
SET SEMI=';'
SET PLUS='+'
SET SINGLE='_'
SET MANY='% '
SET IDQUOTES='`'
SET CURRENCY '$' PREF 2 B
SET NULL ' '
SET DATE FORMAT MM/DD/YYYY
SET DATE SEQUENCE MMDDYY
SET DATE YEAR 30
SET DATE CENTURY 19
CLS
EDIT USING ApplicationMainMenu
RETURN
-- End here ...
```

Notes:

1. Correct the appropriate lines accordingly if you wish to use STATICDB, FASTLOCK, and QUALCOLS settings.
2. Verify the CHARacter SETtings related to your database.
3. Verify the DATE and TIME settings related to your database.
4. Re-visit the routine that closes the application to either EXIT or go back to the Database Explorer or R> prompt.
5. Create for yourself a startup file with SET STATICDB OFF prior to CONNecting the database when you are in "development mode" and making structural changes, when no one else is connected to the same database.

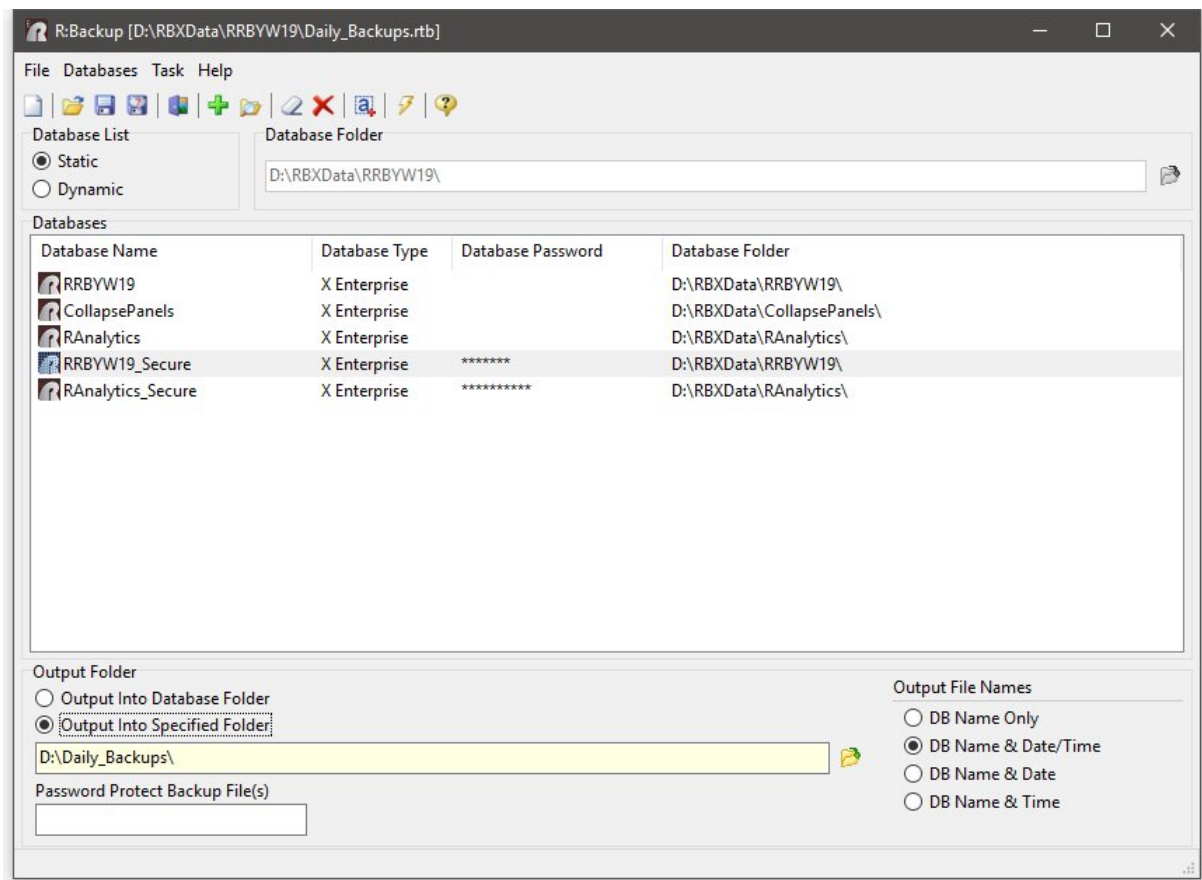
```
-- Example
DISCONNECT
SET STATICDB OFF
CONNECT dbname IDENTIFIED BY ownername
RETURN
```

The options and example detailed above can help you reach maximum performance with R:BASE, but since each application and database is a unique set, only a trial and error process identifies which options help in your situation. Do not forget that different operating systems, hardware configurations, available memory, band-width, system security, and installed anti-virus settings can also affect performance. Experiment with the known options and see which ones improve performance for your application. You, the developer of your application, have the opportunity to set R:BASE internal parameters to gain optimum performance for each application.

1.28 R:Backup Utility

Did you know that you can automate the backup of all your databases using the built-in utility in R:BASE X and R:BASE X Enterprise (Version 10)?

A new utility added to R:BASE X (Version 10) is R:Backup, which is a built-in backup and archive utility for R:BASE databases. Powerful static and dynamic database list options are available to decide what databases can be backed up. When adding files to be backed up, one or more databases can be added from an individual folder. An entire folder, and all sub-folders, can be added to a backup routine.



The backup output location and file names for the R:BASE database files can be controlled with various output options, which allow to specify a folder, or to create the backup files in the same folder as the database. Various "output file" name options allow to specify whether the database name, date, and/or time are used.

R:Backup allows users to save backup projects as script files (.rtb). The script files are used to save backup projects, alter the project settings, and to automatically backup databases as a scheduled task.

Ensuring your databases are safe is easier than ever with R:BASE X and R:BASE X Enterprise!

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.29 R:BASE X or R:BASE X Enterprise?

Designing and deploying version-specific applications that include naming conventions for short (18 character) vs. long (128 character) table, view, column, index, stored procedure, form, label, report, and variable names can be tricky at times to differentiate the deployed versions of R:BASE X/R:BASE X Enterprise.

This is not relevant nor an issue if you are still using the 18 characters naming conventions even in R:BASE X Enterprise. I certainly don't use 18 character naming convention in my vertical market as well as recently deployed government applications running as LAN, WAN, Terminal Server, and Web versions of R:BASE X Enterprise.

So, if you decide to step-up your programming techniques and take advantage of 128 Characters naming convention in R:BASE X Enterprise, while maintaining the similar code for your R:BASE X applications, you may need a method to differentiate the currently running version of R:BASE so that you can use the appropriate form names, etc.

Here is a dynamic CVAL function trick that you can use to find the currently running or compiled version of your R:BASE X/R:BASE X Enterprise.

```
-- Example 01:
IF (SGET(CVAL('VERSION'),10,11)) = 'Enterprise' THEN
  SET NAMEWIDTH 40
  -- additional R:BASE X Enterprise specific code here ...
  EDIT USING MyNewMainMenuForPublicUsers
ELSE
  -- additional R:BASE X specific code here ...
  EDIT USING MainMenuPublicUsrs
ENDIF
-- Example 02:
IF (SGET(CVAL('VERSION'),10,11)) <> 'Enterprise' THEN
  -- additional R:BASE X specific code here ...
  EDIT USING MainMenuPublicUsrs
ELSE
  SET NAMEWIDTH 40
  -- additional R:BASE X Enterprise specific code here ...
  EDIT USING MyNewMainMenuForPublicUsers
ENDIF
```

This should give you an idea of how to implement such techniques in your own applications.

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.30 The Complete R:BASE System Variable List

R:BASE supports the use of system variables that provide information, and enhance the environment, for that R:BASE instance.

Some system variables are preset and constant, some are updated after actions are taken, some are created after actions are taken, while others can be defined to adjust areas of R:BASE.

The existence of following system variables are constant. The values can be verified by typing SHOW VARIABLES at the R> Prompt.

```
R>SHOW VARIABLES
Variable          = Value                                     Type
-----          -
#DATE             = 11/23/2016                                DATE
#TIME             = 03:42 PM                                  TIME
#PI               = 3.14159265358979                          DOUBLE
SQLCODE           = 0                                           INTEGER
SQLSTATE          = 00000                                       TEXT
#NOW              = 11/23/2016 03:42 PM                       DATETIME
```

- #DATE - Stores the current system date
- #TIME - Stores the current system time
- #PI - Stores the value of pi as a DOUBLE data type (3.14159265358979)
- SQLCODE - Holds the result of the previous SQL command
- SQLSTATE - Stores a 5-character long return code string that indicates the status of the previous SQL statement
- #NOW - Stores the current system date and time

The system variables can be used in variable expressions to capture system information.

Examples:

```
SET VAR vToday DATE = .#DATE
SET VAR vTimer TIME = (.#TIME + 10)
SET VAR vNum DOUBLE = (.#PI * 56 ** 2)
SET VAR vRightNow = .#NOW
```

```
R>SHOW VARIABLES
Variable          = Value                                     Type
-----
#DATE             = 11/23/2016                                DATE
#TIME             = 03:42 PM                                  TIME
#PI               = 3.14159265358979                          DOUBLE
SQLCODE           = 0                                           INTEGER
SQLSTATE          = 00000                                      TEXT
#NOW              = 11/23/2016 03:42 PM                       DATETIME
vToday            = 11/23/2016                                DATE
vTimer            = 03:42 PM                                  TIME
vNum              = 9852.03456165758                          DOUBLE
vRightNow         = 11/23/2016 03:42 PM                       DATETIME
```

With the following example;

```
SELECT netamount FROM transmaster +
WHERE netamount IS NULL
IF SQLCODE <> 100 THEN
  --Perform Task here
ENDIF
```

The IF...ENDIF condition checks if any rows exist. If valid rows are found, SQLCODE is set to 0, and the control passes to the command after ENDIF. If no data is found, SQLCODE is set to 100. A negative number for SQLCODE indicates failure. Based on the value of SQLCODE, the WHENEVER command can be used to call an error handling routine.

SQLSTATE was added for Oterro compatibility. Values may be set by attached server tables commands or Stored Procedures.

The following variables can be defined to enhance R:BASE use:

- RBTI_CEM_FONTSIZE - Alters the font size for constraint error messages
- RBTI_CEM_FONTCOLOR - Alters the font color for constraint error messages
- RBTI_ERR_FONTSIZE - Alters the font size for general error messages
- RBTI_ERR_FONTCOLOR - Alters the font color for general error messages
- RBTI_IMG_FOLDER - Defines a valid folder path where all "Open Image" dialogs will initially start
- RBTI_PRNDEFDIR - Specified the default directory for the "Print to File" print dialog
- RBTI_PRNDEFFILE - Specified the default file name for the "Print to File" print dialog
- RBTI_PRNDEFFILET - Specified the default file type for the "Print to File" print dialog

Examples:

```
SET VAR RBTI_CEM_FONTSIZE INTEGER = 14
SET VAR RBTI_CEM_FONTCOLOR TEXT = BLUE
SET VAR RBTI_ERR_FONTSIZE INTEGER = 12
SET VAR RBTI_ERR_FONTCOLOR TEXT = RED
SET VAR RBTI_IMG_FOLDER TEXT = 'C:\TEMP\IMAGES\'
SET VAR RBTI_PRNDEFDIR TEXT = 'C:\Temp\'
SET VAR RBTI_PRNDEFFILE TEXT = 'CSI_Log_'
SET VAR RBTI_PRNDEFFILET TEXT = 'Adobe PDF File'
```

With the FEEDBACK operating condition set ON to display processing results, system variables will be created containing values. The FEEDBACK system variables will be generated and hold the values for the record changes and the elapsed time for the command.

- RBTI_RowsInserted - Holds the value for the number of rows inserted
- RBTI_RowsDeleted - Holds the value for the number of rows deleted
- RBTI_RowsUpdated - Holds the value for the number of rows updated
- RBTI_ElapsedTime - Holds the value for the elapsed processing time

The RBTI_PRNSETUP system variable stores the button selection (OK/CANCEL) after using the PRNSETUP dialog. After launching PRNSETUP window, and pressing "OK", the RBTI_PRNSETUP variable will exist.

```
R>SHOW VARIABLES
Variable                               = Value                               Type
-----
#DATE                                   = 12/21/2012                          DATE
#TIME                                   = 10:43 AM                             TIME
#PI                                      = 3.14159265358979                     DOUBLE
SQLCODE                                  = 0                                      INTEGER
SQLSTATE                                = 00000                                 TEXT
#NOW                                     = 12/21/2012 10:43 AM                  DATETIME
RBTI_PRNSETUP                           = OK                                     TEXT
```

The following system variables are specific to R:BASE Forms:

- RBTI_DBGRID_COLUMN - Stores the name of focused Column on a DB Grid
- RBTI_DIRTY_FLAG - Returns 1 if any database control value(s) in the form was changed, or 0 if nothing was changed
- RBTI_FORM_ALIAS - Stores the name of a focused form, if AS alias is used
- RBTI_FORM_CLICKED - Stores to the clicked control's Component ID property
- RBTI_FORM_COLNAME - Stores the name of focused column of a database control on a form
- RBTI_FORM_COLVALUE - Stores the value of focused column database control on a form
- RBTI_FORM_COMPID - Stores the value of focused DB/Variable Edit control's Component ID
- RBTI_FORM_CTRLTEXT - Stores to the clicked control's "Caption" or "Text" property
- RBTI_FORM_DATATYPE - Stores the data type of focused column database control on form
- RBTI_FORM_DBLCLKED - Stores the double clicked control's Component ID property
- RBTI_FORM_DSGN - Stores the value of "YES" when a form is open in the Form Designer
- RBTI_FORM_DIRTYVAR - Returns 1 if any Variable Control value(s) in form were changed, or 0 if nothing was changed
- RBTI_FORM_EXT_DSGN - holds the value of "YES" when a form is open in the External Form Designer
- RBTI_FORM_FORMNAME - Stores the name of the current form (useful when using multiple forms simultaneously)
- RBTI_FORM_MODE - Stores the value of current mode of the form; ENTER, EDIT, or BROWSE
- RBTI_FORM_TBLNAME - Stores the name of the current table in a form (useful with multi-table forms)
- RBTI_FORM_VARNAME - Stores the name of focused Variable Control on a form
- RBTI_FORM_VARVALUE - Stores the value of focused Variable Control on a form

The following system variable is specific to R:BASE Reports:

- RBTI_REPORT_NAME - Stores the name of the current report

The following system variable is specific to R:BASE Labels:

- RBTI_LABEL_NAME - Stores the name of the current label

When using the Form Designer, the "System Variable" control is an object that can be placed on the form to display system information. The following values are available:

- Date
- Date and Time
- Time
- R:BASE Name
- R:BASE User

- Windows User
- Computer Name
- Database Name
- Default Printer
- Current Printer
- R:BASE Version
- R:BASE Build
- R:BASE Version and Build
- Current Directory
- Record Count
- Record Number and Count
- Record Number and Count with Description
- Record Number
- Record Number with Description

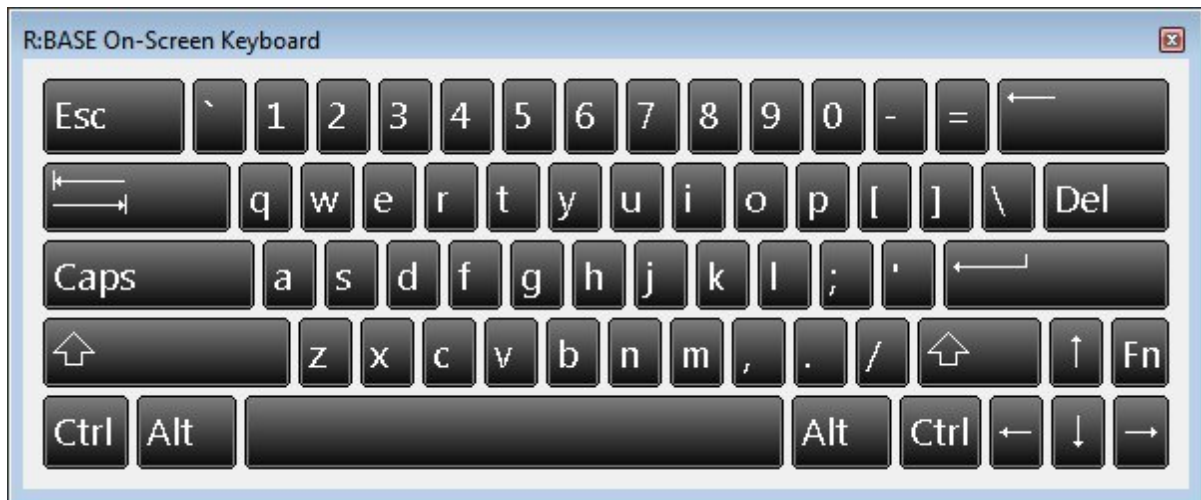
When using the Report/Label Designer, the "System Variable" control is an object that can be placed on the report/label to display system information. The following values are available:

- Date
- Date and Time
- Document Name
- Print Date/Time
- Page Count
- Page Set
- Page Set with Description
- Page Number
- Page Number with Description
- Time
- R:BASE Name
- R:BASE User
- Windows User
- Computer Name
- Database Name
- Default Printer
- Current Printer
- Version
- Build
- Version/Build
- Current Folder

1.31 The R:BASE On-Screen Keyboard (ROSK)

Did you know that you can display an on-screen keyboard within R:BASE to allow users to type and enter data, without relying on a physical keyboard?

Imagine your mobile R:BASE application running on Tablet PCs in a industrial, manufacturing, or austere environments. The R:BASE On-Screen Keyboard is a visual keyboard displayed on the screen that provides all standard keys, where users can select keys using the mouse or another pointing device.



The R:BASE On-Screen Keyboard is the ideal complement to any Handheld, Kiosk, Mobile, Pen, Rugged, Tablet or Touchscreen based R:BASE eXtreme applications.

The R:BASE On-Screen Keyboard (ROSK) is accessible using the following methods:

- Selecting "Utilities" > "On Screen Keyboard" from the main Menu Bar
- Pressing the [Ctrl]+[Alt]+K key combination
- Using the PROPERTY command

Important Note: The R:BASE On-Screen Keyboard is only supported on Windows 7, Windows 2008 Server R2, and higher operating systems.

Supported PROPERTY Commands:

--To show/hide the keyboard

```
PROPERTY ROSK SHOW TRUE/FALSE
```

--To alter the color of the button outer edge:

```
PROPERTY ROSK OUTER_EDGE_COLOR_NORMAL Value
PROPERTY ROSK OUTER_EDGE_COLOR_PRESSED Value
PROPERTY ROSK OUTER_EDGE_COLOR_DISABLED Value
```

--To alter the color of the button inner edge:

```
PROPERTY ROSK INNER_EDGE_COLOR_NORMAL Value
PROPERTY ROSK INNER_EDGE_COLOR_PRESSED Value
PROPERTY ROSK INNER_EDGE_COLOR_DISABLED Value
```

--To alter the starting gradient color of the button background:

```
PROPERTY ROSK BK_START_COLOR_NORMAL Value
PROPERTY ROSK BK_START_COLOR_PRESSED Value
PROPERTY ROSK BK_START_COLOR_DISABLED Value
```

--To alter the ending gradient color of the button background:

```
PROPERTY ROSK BK_END_COLOR_NORMAL Value
PROPERTY ROSK BK_END_COLOR_PRESSED Value
PROPERTY ROSK BK_END_COLOR_DISABLED Value
```

--To alter the font colors of the buttons:

```
PROPERTY ROSK TEXT_COLOR_NORMAL Value
PROPERTY ROSK TEXT_COLOR_PRESSED Value
PROPERTY ROSK TEXT_COLOR_DISABLED Value
```

--To alter the style of the buttons:

```
PROPERTY ROSK DRAWING_STYLE NORMAL / GRADIENT
PROPERTY ROSK GRADIENT_START_COLOR Value
PROPERTY ROSK GRADIENT_END_COLOR Value
```

Example:

```
PROPERTY ROSK SHOW TRUE
PROPERTY ROSK BK_START_COLOR_NORMAL WHITE
PROPERTY ROSK BK_END_COLOR_NORMAL RED
PROPERTY ROSK BK_START_COLOR_PRESSED white
PROPERTY ROSK BK_END_COLOR_PRESSED GRAY
PROPERTY ROSK OUTER_EDGE_COLOR_NORMAL BLACK
PROPERTY ROSK OUTER_EDGE_COLOR_PRESSED RED
PROPERTY ROSK INNER_EDGE_COLOR_NORMAL WHITE
PROPERTY ROSK INNER_EDGE_COLOR_PRESSED RED
PROPERTY ROSK TEXT_COLOR_NORMAL BLACK
PROPERTY ROSK TEXT_COLOR_PRESSED BLACK
PROPERTY ROSK DRAWING_STYLE GRADIENT
PROPERTY ROSK GRADIENT_START_COLOR BLUE
PROPERTY ROSK GRADIENT_END_COLOR WHITE
```



Have fun adding the R:BASE On-Screen Keyboard to your custom applications!

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.32 Tracking Date/Time and Version Information

Did you know that you can track the Date/Time and Version information of Stored Procedures, Forms, Reports, and Labels in R:BASE X Enterprise using the Database Explorer?

In addition to SYS_PROC_VERSION (INTEGER) & SYS_PROC_MOD_TS (DATETIME) for SYS_PROCEDURES, R:BASE eXtreme (64) now includes the following additional columns for SYS_FORMS3, SYS_LABELS3, and SYS_REPORTS3 to keep track of date/time as well as the revision number of modified Forms, Labels, and Reports.

Additional SYS_ Columns:

- SYS_FORMS3 (SYS_FORM_MOD_TS, SYS_FORM_VERSION)
- SYS_LABELS3 (SYS_LABEL_MOD_TS, SYS_LABEL_VERSION)

- SYS_REPORTS3 (SYS_REPORT_MOD_TS, SYS_REPORT_VERSION)

These columns are updated when saving the form, label or report while in form, label or report designers accordingly, which are displayed at the Database Explorer.

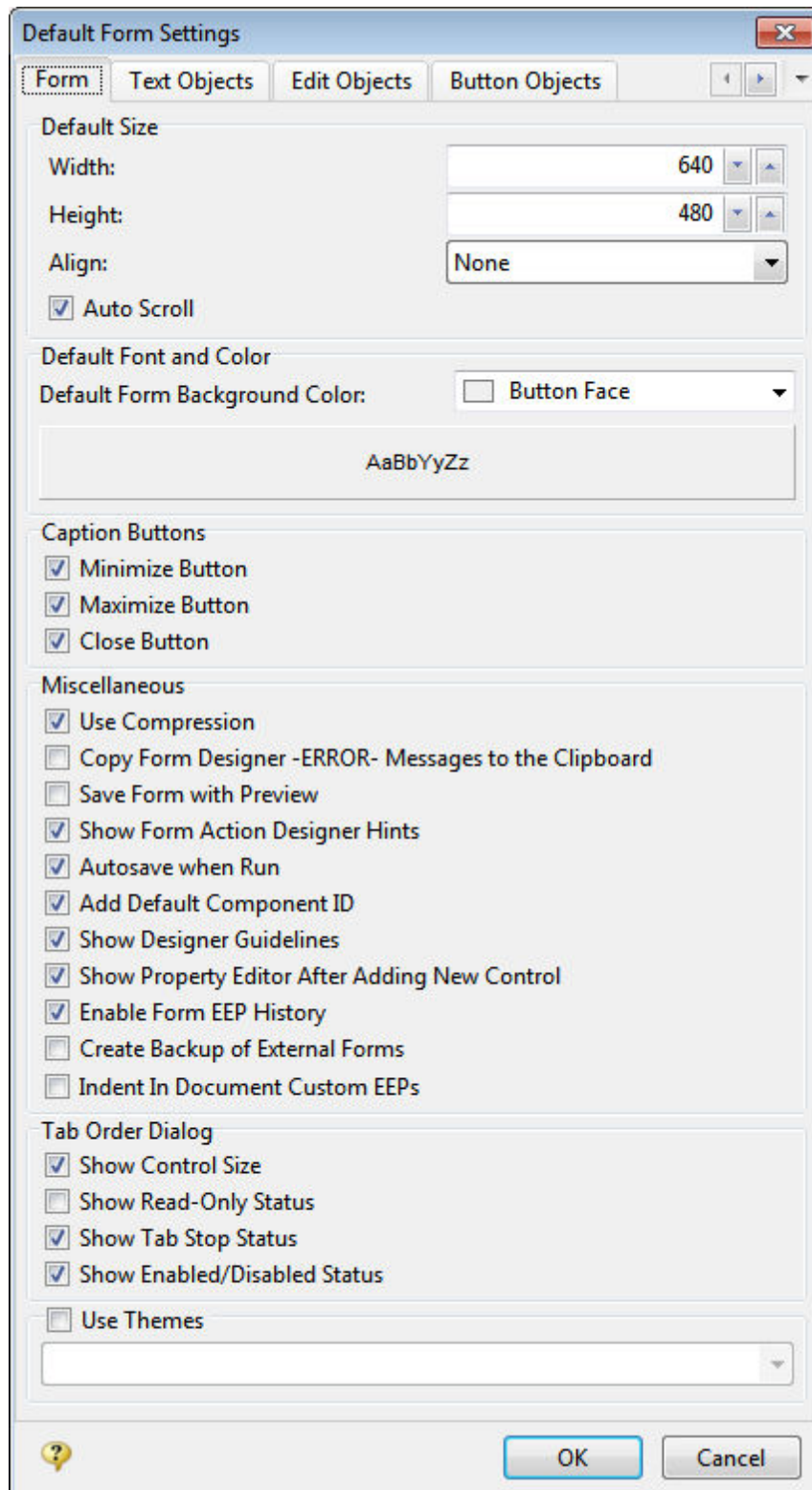
Report Name ▲	Report Table(s)	Comment	Size (bytes)	Date/Time	Version
1. AdvancedRichText	Contact	Advanced Rich Text Report	44,687	11/23/2016 15:57:00	2
2. ColumnarReport	Customer	To Demonstrate Columnar Report	3,923	11/23/2016 15:57:00	1
3. ConditionalChkBox	YTDInvoiceTotal	To Demonstrate the use of Conditional CheckBox Pro	10,385	11/23/2016 15:57:00	1
4. Contacts	Contact	Customer Contact List	6,255	11/23/2016 15:57:00	1
5. ContactsByCustomer	Contact	Contact List by Customer	7,223	11/23/2016 15:57:00	1
6. CoverPageAndDetail	Customer	Customer List by State (Summary Page with Details)	8,509	11/23/2016 15:57:00	1
7. CrossTabReport	TCrossTabView	CrossTab Report (Standard)	6,165	11/23/2016 15:57:00	1
8. CrossTabReportCust	TCrossTabView	CrossTab Report (Custom)	6,183	11/23/2016 15:57:00	1
9. CustContInvoices	NewCustomer (New	Sub-Reports (4 Tables): NewCustomer, NewContact, I	19,898	11/23/2016 15:57:00	1
10. CustDirWithURL_DB	Customer	Customer Phone Directory with Hyperlinks (URLs)	5,613	11/23/2016 15:57:00	1
11. CustDirWithURL_VAR	Customer	Customer Phone Directory with Hyperlinks (URLs)	5,543	11/23/2016 15:57:00	1
12. CustomerContacts	Customer (Contact)	Sub-Reports (2 Tables): Customer and Contacts	9,296	11/23/2016 15:57:00	1
13. CustomerDirectory	Customer	Customer Phone Directory	5,449	11/23/2016 15:57:00	1
14. CustomerList	Customer	List of Active Customers	6,104	11/23/2016 15:57:00	1
15. Customers	Customer	Master List of Customers	6,238	11/23/2016 15:57:00	1
16. CustomFormLetter	Customer	Custom Form Letter Using Advanced Rich Text Contr	14,784	11/23/2016 15:57:00	1
17. CustomPageInvoice	ProdAlias	Invoice with Custom Page Numbers	21,836	11/23/2016 15:57:00	1
18. CustYellowPages	Customer	Customer Phone Directory (Yellow Pages)	5,409	11/23/2016 15:57:00	1
19. DBCalcLineCount	SalesDataByCompar	YTD Sales Summary and Sub-Totals by Customer - Lir	14,876	11/23/2016 15:57:00	1

Label Name ▲	Main Table	Comment	Size (bytes)	Date/Time	Version
1. ContactLabel	Contact	Contact Address Label	2,602	11/23/2016 15:57:00	1
2. CustomerLabels	Customer	Customer Labels	2,349	11/23/2016 15:57:00	1
3. CustShippingLabel	Customer	Customer Shipping Label	3,926	11/23/2016 15:57:00	1
4. CustShpLabl	Customer	2DBarcode Shipping Label	3,945	11/23/2016 15:57:00	1
5. EmployeeLabel	Employee	Employee Labels	2,878	11/23/2016 15:57:00	1
6. EmployeeLabel_01	Employee	Employee Labels	2,800	11/23/2016 15:57:00	1
7. EmployeeNameTags	Employee	Employee Name Tags	2,231	11/23/2016 15:57:00	1
8. LabelExpressions	Customer	Customer Labels	1,945	11/23/2016 15:57:00	1
9. ProductLabels	Product	Product Labels with Bar Code	3,159	11/23/2016 15:57:00	1
10. SecureNmTag	Employee	2DBarcode Security Tag	2,166	11/23/2016 15:57:00	1
11. SecurityNameTags	Employee	Employee Name Tags	2,231	11/23/2016 15:57:00	1

Hint:

While in Form Designer, when running or previewing the form, the form is automatically saved, thus stamps the date/time as well as increases the version count. If you wish to change this behavior and only stamp the date/time and increase the version count when you click on the [Save] button in Form Designer, you may change the Form Designer settings as follows:

Database Explorer | Main Menu | Settings | Form Designer ...



Supported Versions:

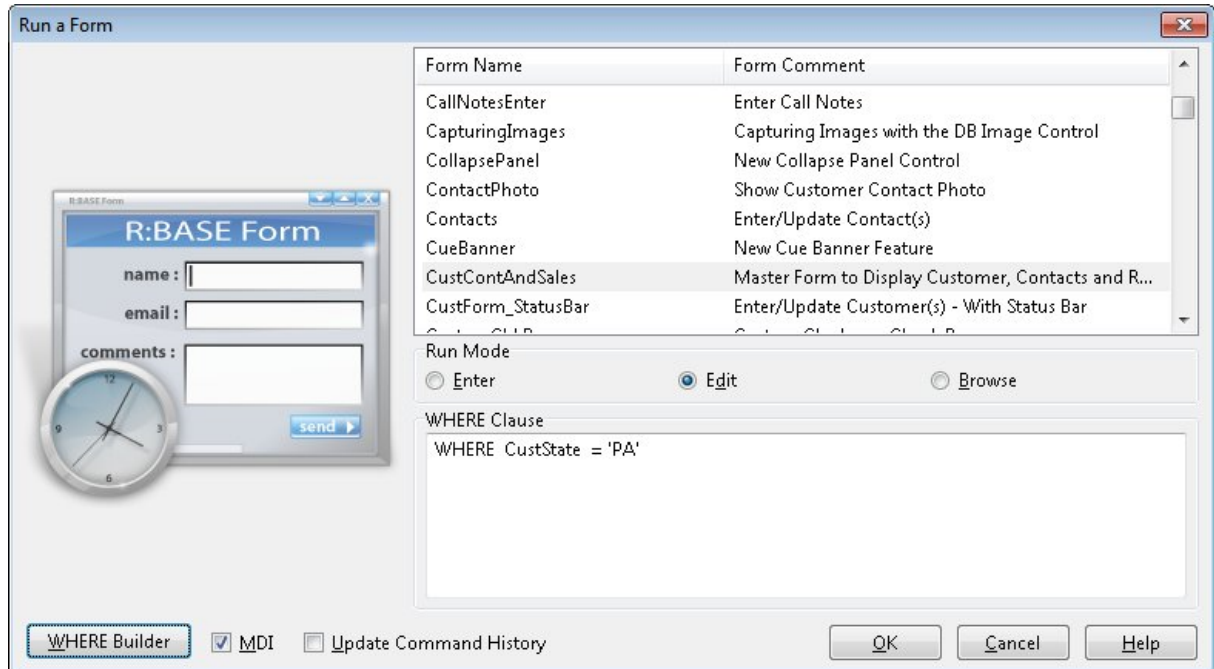
- R:BASE X (Build...: 10.0.1.10522 or higher)

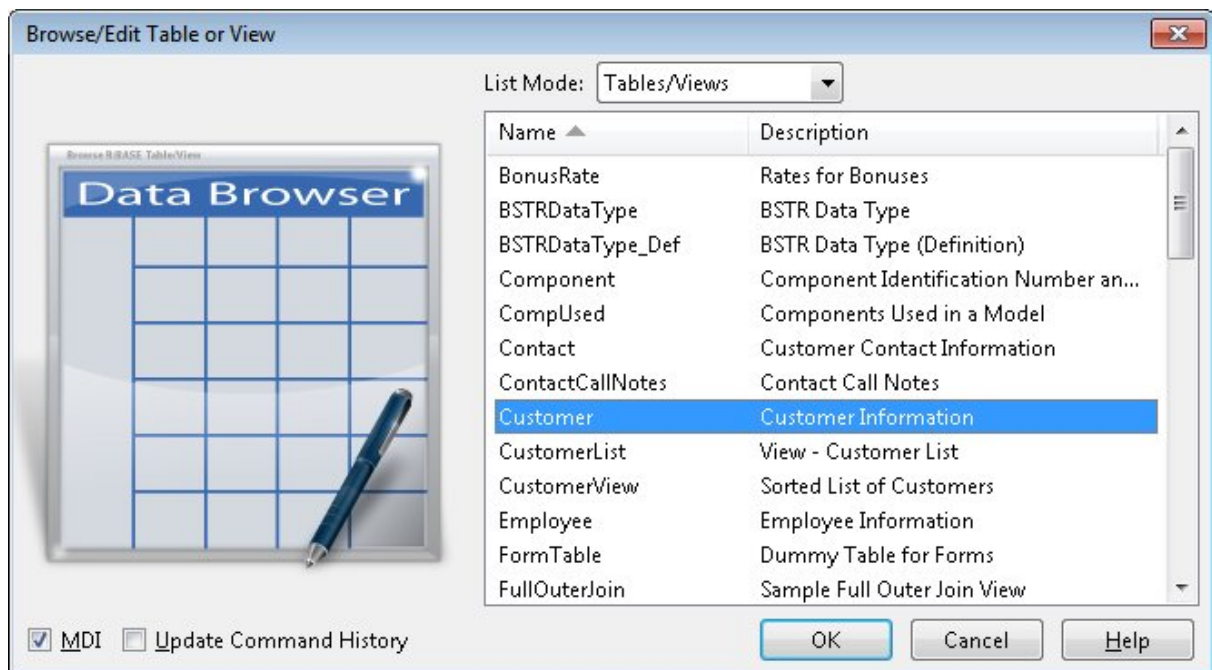
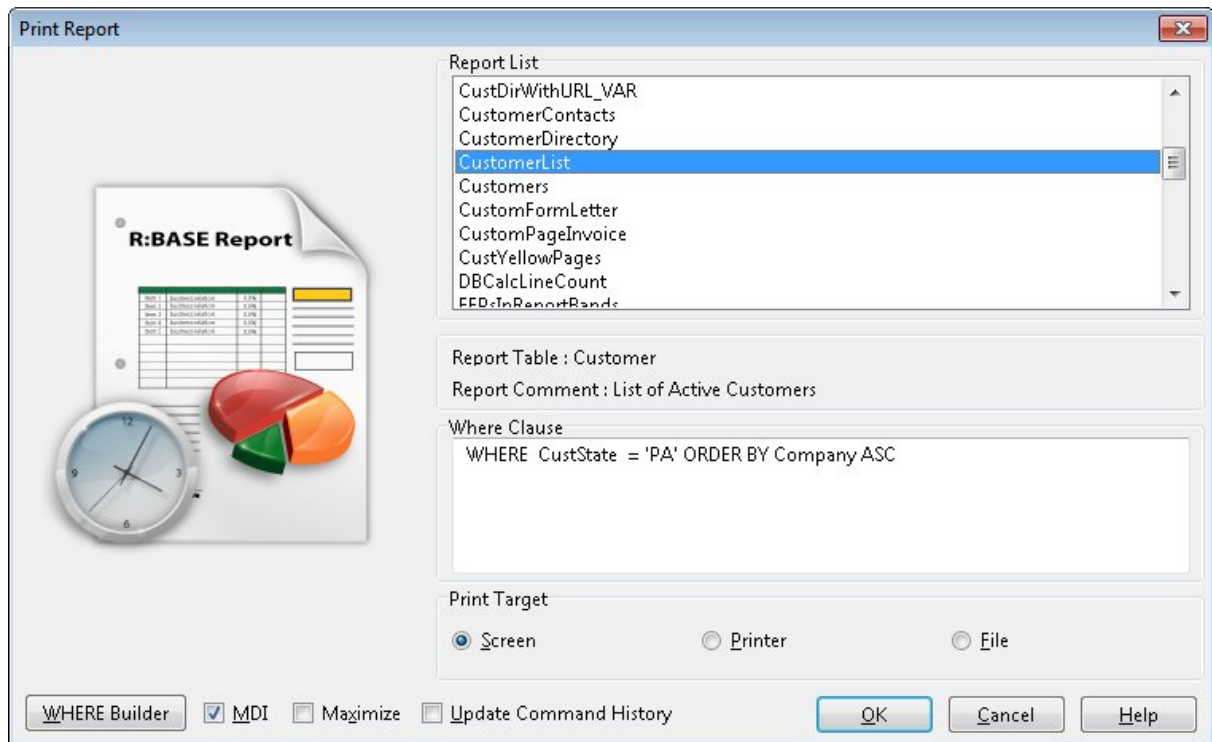
1.33 Update Command History

Did you know that you can update the R> Prompt Command History tool bar when running a form, printing a report, browsing a table, or when building a clause within the WHERE Builder?

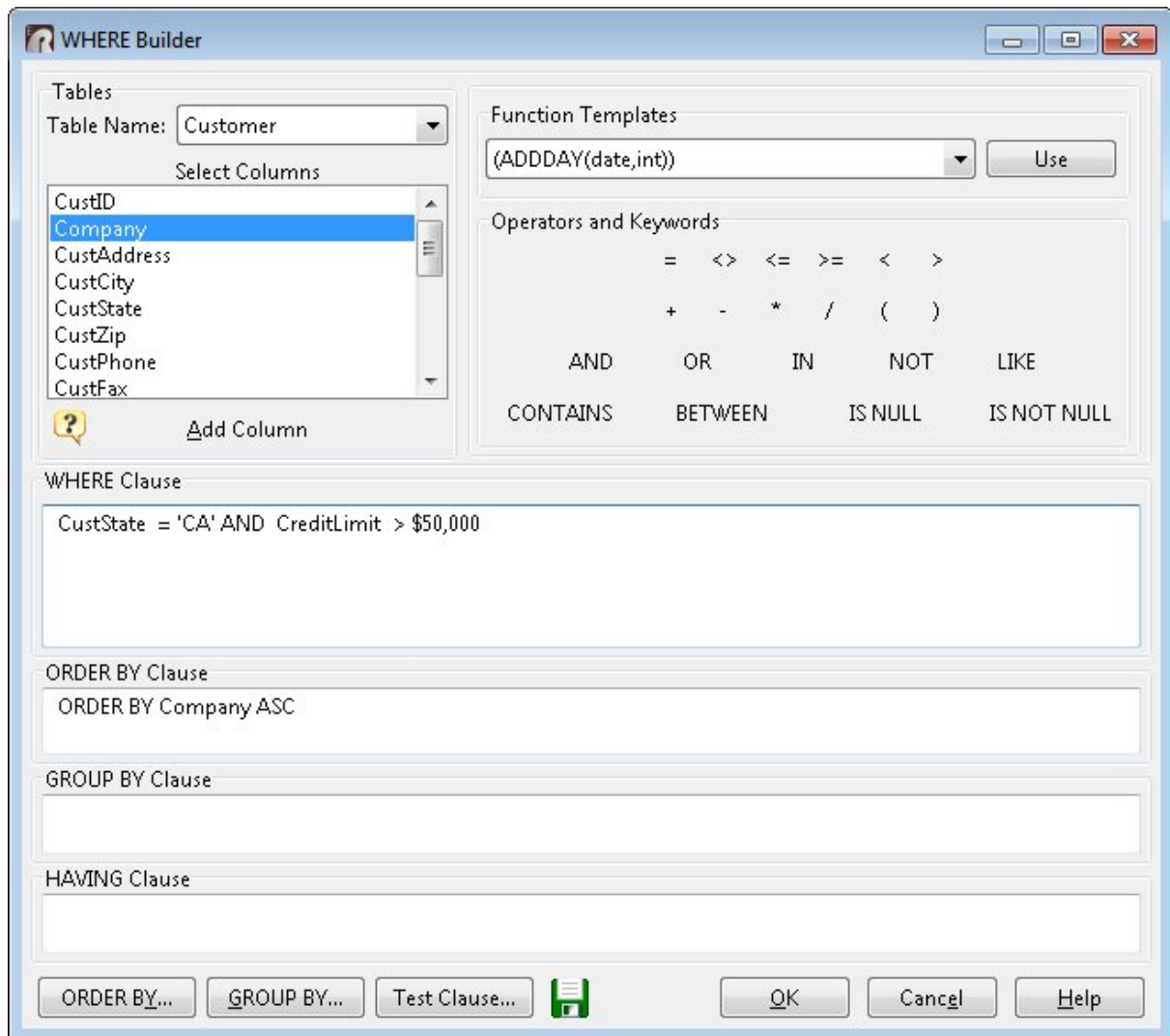
Within any of the above areas, the syntax is executed through a GUI, when using the full version of R:BASE. An added option is now available to update the R> Prompt Command History, allowing end-users or developers to easily reference the command syntax.

A "Update Command History" check box is visible within the following:





A green diskette button is visible within the Where Builder:



Supported Versions:

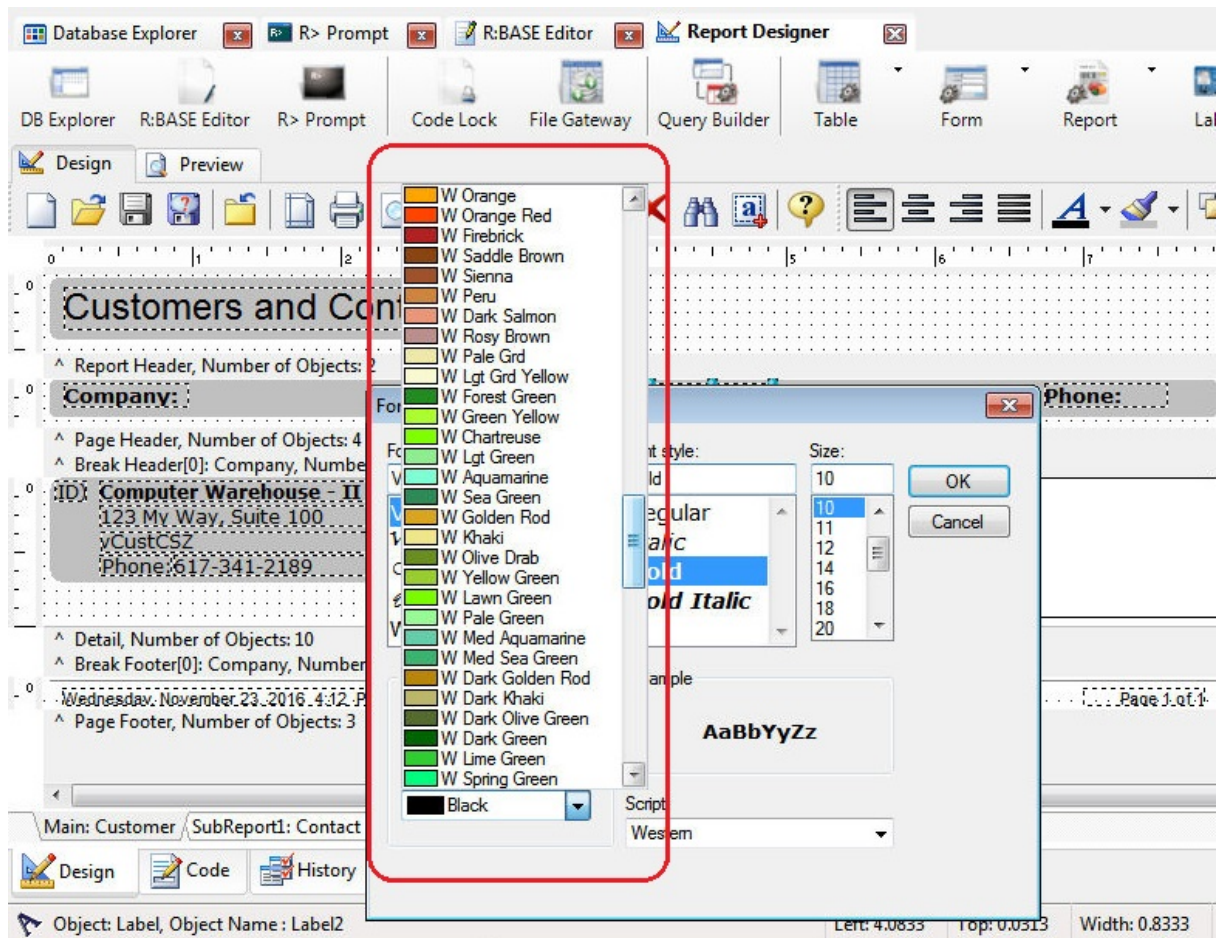
- R:BASE X (Build..: 10.0.1.10522 or higher)

1.34 Using Expanded Color Palette

The Data Dictionary [F3] Key in R:BASE provides the option to choose a color from the list of expanded colors.

While programming and customizing applications, if you wish to use custom font color, background color, for CHOOSE, DIALOG, or PAUSE commands you can press the [F3] key to open the Data Dictionary, select the color and then either double-click for single item or [OK] button for multi-select items.

Did you know that now you can select all 188 colors from the "Font" Control Panel while in the Form, Label, or Report Designers?



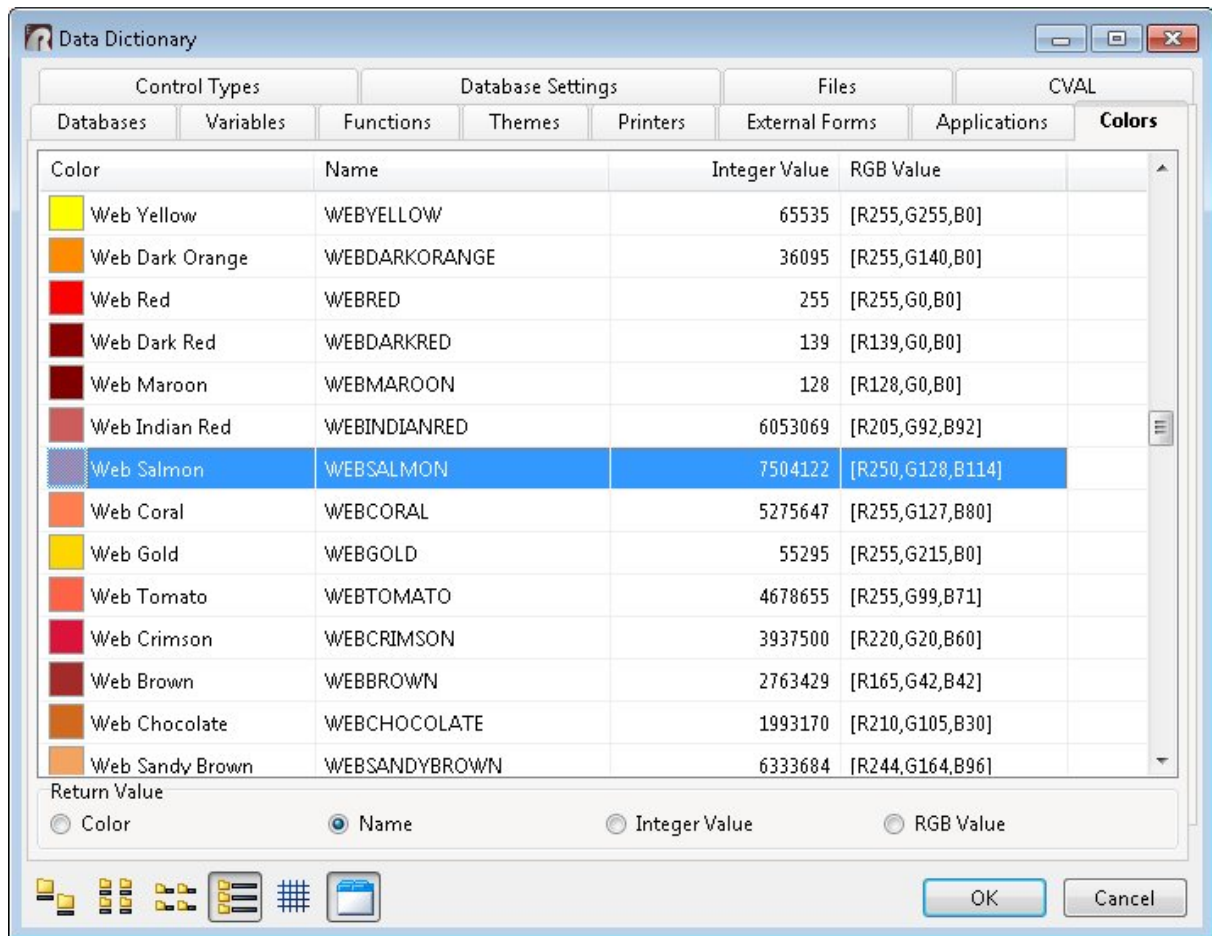
Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.35 Using Extended Colors

Did you know that you can use the extended colors defined in R:BASE eXtreme that can be accessed while in Form, Label or Report Designers?

You can also use enhanced Data Dictionary [F3] key to access the complete list of colors along with Displayed Color, Name, Integer and RGB value.



Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.36 Using Foreign Keys in TEMPORARY Tables

Since the introduction of TEMPORARY Tables and Views in July 1997 (R:BASE 6.1), and the constant innovations in the latest versions, the use of TEMPORARY tables & views has become the most valuable feature in R:BASE.

While using TEMPORARY tables and collecting data, the referential integrity becomes a challenge. Did you know that now you can define a FOREIGN KEY on a TEMPORARY table having REFERENCES to a permanent table?

The latest versions of R:BASE now support the use of adding foreign keys (FK) on a TEMPORARY table with REFERENCES to a permanent table.

R:BASE Data Designer (RBDFine) also supports the ability to add FOREIGN KEY constraints to TEMPORARY tables.

Here is sample code that defines TEMPORARY Tables with PRIMARY and FOREIGN KEYS (based on RRBYW19 sample database).

```
-- Example 01
-- Define TEMPORARY tInvoiceHeader Table
SET ERROR MESSAGE 2038 OFF
DROP TABLE tInvoiceHeader
```

```

SET ERROR MESSAGE 2038 ON
CREATE TEMPORARY TABLE `tInvoiceHeader` +
(`TransID` INTEGER NOT NULL +
('Value for column TransID cannot be NULL'), +
`CustID` INTEGER, +
`EmpID` INTEGER, +
`TransDate` DATE, +
`BillToCompany` TEXT (40), +
`BillToAddress` TEXT (30), +
`BillToCity` TEXT (20), +
`BillToState` TEXT (2), +
`BillToZip` TEXT (10), +
`ShipToCompany` TEXT (40), +
`ShipToAddress` TEXT (30), +
`ShipToCity` TEXT (20), +
`ShipToState` TEXT (2), +
`ShipToZip` TEXT (10), +
`NetAmount` CURRENCY, +
`Freight`= (NetAmount*.01) CURRENCY, +
`Tax`= (NetAmount*.081) CURRENCY, +
`InvoiceTotal`= (NetAmount+Freight+Tax) CURRENCY)
ALTER TABLE `tInvoiceHeader` ADD PRIMARY KEY +
(`TransID`) +
('Values for rows in tInvoiceHeader must be Unique',+
'Cannot Delete - values exist in another table',+
'Cannot Update - values exist in another table')
ALTER TABLE `tInvoiceHeader` ADD FOREIGN KEY +
(`CustID`)+
REFERENCES `Customer` +
('Cannot Insert - value does not exist in Customer',+
'Cannot Update - value does not exist in Customer')
COMMENT ON TABLE `tInvoiceHeader` IS +
'TEMPORARY Table for Invoice Header'
RETURN

-- Example 02
-- Define TEMPORARY tInvoiceDetail Table
SET ERROR MESSAGE 2038 OFF
DROP TABLE tInvoiceDetail
SET ERROR MESSAGE 2038 ON
CREATE TEMPORARY TABLE `tInvoiceDetail` +
(`TransID` INTEGER, +
`DetailNum` INTEGER, +
`Model` TEXT (6), +
`Units` INTEGER, +
`Price` CURRENCY, +
`Discount` REAL, +
`SalePrice`= +
(Price- (Price*Discount/100)) CURRENCY, +
`ExtPrice`= (Units*SalePrice) CURRENCY)
ALTER TABLE `tInvoiceDetail` ADD FOREIGN KEY +
(`TransID`)+
REFERENCES `tInvoiceHeader` +

```

```
('Cannot Insert - value does not exist in tInvoiceHeader',+
'Cannot Update - value does not exist in tInvoiceHeader')
COMMENT ON TABLE `tInvoiceDetail` IS +
'TEMPORARY Invoice Detail Table'
RETURN
```

1.37 Using R:BASE Executable Startup Options

R:BASE for Windows supports several startup parameters that users can embed in the desktop shortcut properties, or use as command line switches, to launch a custom R:BASE application, or to launch R:BASE in a different folder. When multiple parameters are specified, each must be separated with a space.

Target:

-A

This option tells R:BASE to look in the specified "Target" folder first for the R:BASE engine and configuration files. This option is beneficial if you are making only one installation of R:BASE on a network server and want local workstations to be able to launch the R:BASE X program without the need to install it on the local workstation. Add the "-A" parameter after the executable at the end of the "Target:" field value. The executable and "-A" parameter should be separated with a space.

"-BLOB <filename>"

This option opens the specified file (rich text, image, etc.) in the R:BASE BLOB Editor as the initial screen, when the program is launched. Note that there is a single space between the -BLOB and the file name parameter, and that the -BLOB <filename> must be enveloped in double-quotes. Example: C:\RBTI\RBGXEXE\RBGXEXE.EXE "-BLOB C:\RBTI\RBGXEXE\License.rtf"

-C or -P

This option opens the R> Prompt window when the program is launched.

-E

This option opens the R:BASE Editor when the program is launched.

"-E <filename>"

This option opens the specified file in the appropriate file editor when the program is launched. An RFF file is opened in the External Form Designer, an RBA file in the Application Designer, and the rest in the internal R:BASE Editor. Note that there is a single space between the -E and the file name parameter, and that the -E <filename> must be enveloped in double-quotes. Example: C:\RBTI\RBGXEXE\RBGXEXE.EXE "-E C:\RBTI\RBGXEXE\ReadMe.txt"

-FS

This option forces the display of splash screen, even if it is set to not be shown in the Startup Options, or if "-S" is used

-L

This option opens the Database Explorer when the program is launched.

"-O filespec"

Specifies an alternate R:BASE configuration file to use for startup information. Example: C:\RBTI\RBGXEXE\RBGXEXE.EXE "-O C:\Users\Steve\Documents\RBENGINEXE.CFG"

filespec

This option specifies a startup file opens your custom application directly. It can be a command file (.DAT, .RMD, .CMD) or an R:BASE application file (.RBA).

-S

This option opens R:BASE in silent mode, where the splash screen is hidden even if it is set to be shown in the Startup Options.

-Txxx

This option adds an additional text label to an R:BASE instance, where xxx is a custom text or "RandomID", to better identify the running the instance in the Task Manager. This is useful when running multiple instances of R:BASE. A recommended parameter is "-TRandomID" to use the current operating

system TickCount value, which returns the number of milliseconds elapsed since the system started.
Example: C:\RBTI\RBGX\RBGX.EXE -TRandomID

-X

This option does not load the Database Explorer, R> Prompt, or R:BASE Editor, even if the modules are set to be shown in the Startup Options. This parameter has a lower priority than -C, -P, -L, and -E. For example, the editor will be shown if both -X and -E is used. This option is useful when R:BASE is used as a file editor, as the program can be loaded with only the R:BASE Editor displayed at startup.

-RBACKUP

This option specifies R:BASE will open as only as R:Backup. When this switch is present, other switches meant for R:BASE are ignored. R:Backup specific parameters include the following:

-S

Runs R:Backup in silent mode

-R

Runs an R:Backup project

filespec

Specifies the R:Backup file (.rtb) to be loaded or executed

Start in:

By editing the "Start in:" field, you can alter what folder location R:BASE will launch in. You can specify a local folder, a mapped network drive, or a UNC path.

Launching Applications

To launch a custom application on a network folder, edit the "Start in:" field to the database and custom application folder. Then, specify your R:BASE application file or startup file in the "Target:" after the R:BASE executable. Separate the executable and startup file with a space.

Notes:

- The order in which the parameters are listed in the Target field is the order in which the modules will open and appear in the R:BASE program, with the last parameter displayed. For example, using the following Target: C:\RBTI\RBGX\RBGX.EXE -E -L -C , R:BASE will open with the modules in the order of R:BASE Editor, Database Explorer, and R> Prompt, with the R> Prompt window displayed.
- When multiple parameters are specified, each must be separated with a space.

Examples

Example 01:

Launches R:BASE X Enterprise in the RRBYW19 sample database folder, opening the R:BASE Editor, R> Prompt, and Database Explorer modules while displaying the Database Explorer.

Target: C:\RBTI\RBGX\RBGX.exe -E -C -L
Start in: C:\RBTI\RBGX\Samples\RRBYW19

Example 02:

Launches R:BASE X in the program folder, opening the Database Explorer and R:BASE Editor modules, while displaying R:BASE ReadMe.txt text file.

Target: C:\RBTI\RBGX\RBGX.exe -L "-E C:\RBTI\RBGX\ReadMe.txt"
Start in: C:\RBTI\RBGX

Example 03:

Launches R:BASE X Enterprise in the program folder, opening the Database Explorer and R> Prompt modules, while initially displaying R:BASE License.rtf in the BLOB Editor.

Target: C:\RBTI\RBGX\RBGX.exe -L -C "-BLOB C:\RBTI\RBGX\License.rtf"
Start in: C:\RBTI\RBGX

Example 04:

Launches locally installed R:BASE X in a shared network folder, and launches the CRM.DAT startup file.

```
Target: C:\RBTI\RBGX\RBGX.exe CRM.DAT
Start in: R:\RBFILERS\CRM
```

Example 05:

Launches a network installation of R:BASE X Enterprise in a shared network folder with the -A parameter, and launches the SPECS.DAT startup file.

```
Target: R:\RBTI\RBGX\RBGX.exe -A SPECS.DAT
Start in: R:\RBFILERS\SPECS
```

Example 06:

Launches a network installation of R:BASE X in a shared network folder with the -A parameter, uses a custom configuration file for the user, and launches the EST.DAT startup file.

```
Target: R:\RBGX\RBGX.exe -A "-O R:\RBGX\CFG\JOE\RBENGINEX.CFG" EST.DAT
Start in: R:\RBFILERS\EST
```

Example 07:

Launches R:BASE X Enterprise in a development folder on the D:\ drive, with a custom configuration file, and uses a random ID to identify several R:BASE instances within the Windows Task Manager.

```
Target: C:\RBTI\RBGX\RBGX.exe "-O C:\Users\Glen\Documents\RBENGINEX.CFG" -TRandomID
Start in: D:\RBASEDEV
```

Supported Versions:

- R:BASE X (Build..: 10.0.1.10522 or higher)

1.38 Using the #NOW System Variable

Did you know that in addition to #DATE, #TIME, #PI, SQLCODE, and SQLSTATE, now you also have an additional #NOW system variable?

Example 01:

```
SHOW VARIABLES
```

Among other system variables, such as #DATE, #TIME, #PI, SQLCODE, and SQLSTATE, #NOW will also be included when using the SHOW VARIABLES command at the R> prompt.

Example 02:

```
SET VAR vCurrentDateTime = .#NOW
```

The resulting variable vCurrentDateTime will return current date time with DATETIME data type.

Example 03:

```
UPDATE tablename SET datetimetype = .#NOW WHERE ...
```

Will update the datetimetype with DATETIME data type with current date and time value.

Example 04:

```
INSERT INTO tablename (Colname1, Colname2, Colname3) +
VALUES .vColumnValue1, .vColumnValue2, .#NOW
```

1.39 Using the Enhanced Find in Files Utility

Did you know that you can open the Find in Files utility and it will now load with its previously used options, along with a combo box with the last 10 searched items?

Using the built-in Editor in R:BASE X/R:BASE X Enterprise or the stand-alone R:BASE Editor X, the "Find in Files" option will now remember its previously used options, along with a combo box with the last 10 searched items.

In addition, the "Skip Files" field has been added to the Find in Files search utility to exclude files from the text search.

1.40 Using the Magnifying Glass

Did you know that you can use the Magnifying Glass to zoom any object, text, or even Expression Window when defining a complex expression while in Form, Label, or Report Designer? R:BASE includes an option to use a cool Magnifying Glass on demand.

To display the Magnifying Glass in R:BASE, select "Utilities" > "Magnifying Glass" from the Menu Bar.

Form Name ▲	Main Table	Comment	Size (bytes)	Date/Time	Version ▲
1. ActiveX	FormTable	To Demonstrate the use of ActiveX	1,178	10/09/2014 16:40:50	1
2. AdhocReports		Ad hoc Reports	14,985	10/09/2014 16:40:50	1
3. AnimatedGIFs	FormTable	To Demonstrate the use of Animated GIFs	73,391	10/09/2014 16:40:50	1
4. AskScreenPrinter	FormTable	Options to Print Report on Screen or Printer	4,415	10/09/2014 16:40:50	1
5. BalloonTip		New Balloon Tip Feature	8,534	10/21/2014 11:43:52	4
6. CallNotesEdit	ContactCallNotes	Review/Update Call Notes	8,972	10/09/2014 22:44:53	5
7. CallNotesEnter	ContactCallNotes	Enter Call Notes	4,752	10/09/2014 16:40:50	1
8. CapturingImages	Employee	Capturing Images with the DB Image Control	8,059	10/09/2014 16:40:50	1
9. CollapsePanel		New Collapse Panel Control	4,901	10/09/2014 16:40:50	1
10. ContactPhoto	Contact	Show Customer Contact Photo	2,334	10/09/2014 16:40:50	1
11. Contacts			7,113	10/09/2014 16:40:50	1
12. CueBanner			6,844	10/09/2014 16:40:50	1
13. CustContAndSales			12,756	10/09/2014 16:40:50	1
14. CustForm_StatusBar			8,011	10/09/2014 16:40:50	1
15. CustomChkBox			7,033	10/09/2014 16:40:50	1
16. CustomerContacts			11,788	10/09/2014 16:40:50	1
17. CustomerManagement			6,837	10/09/2014 16:40:50	1
18. Customers			7,719	10/09/2014 16:40:50	1
19. Customers_Users	Customer	Enter/Update Customer(s)	10,604	10/09/2014 16:40:50	1
20. CustomFormActions	FormTable	To Demonstrate the Use of Custom Form Ac	3,647	10/09/2014 16:40:50	1
21. CustomPopupMenus	FormTable	To Demonstrate the use of Custom Popup M	6,697	10/09/2014 16:40:50	1
22. CustomPopupSplash	FormTable	Custom Pop-ups Splash Screen	1,386	10/09/2014 16:40:50	1
23. CustomSpeedBtns		Custom Glyphs on Speed Buttons	17,160	10/09/2014 16:40:50	1
24. CustSearchByST	FormTable	Customer Search By State (Using MAP Imag	93,614	10/09/2014 16:40:50	1
25. CustSearchByST_Tot	FormTable	Customer Search By State (Using MAP Imag	96,674	10/09/2014 16:40:50	1
26. CVALFunctions		New CVAL Functions	6,908	10/09/2014 16:40:50	1
27. DateTimePicker		New Date/Time Picker Options	4,904	10/09/2014 16:40:50	1
28. DBChkBoxInSR	Customer	DB CheckBox in Scrolling Region	8,370	10/09/2014 16:40:50	1
29. DBGridProperties	Employee	To Demonstrate the Use of Dynamically Ch	2,023	10/09/2014 16:40:50	1

Did you know that you can expand the horizontal and vertical size of the Magnifying Glass by grabbing the corner or edge of the transparent glass displayed at the center of the screen?

Tips:

- [Ctrl]+[Alt]+M will launch the built-in Magnifying Glass (anywhere)
- [Esc] or "Right-Click | Close" will close the Magnifying Glass
- Right-Click will provide the option to zoom (2x, 4x, 8x, 16x)

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

1.41 Using the Scratch Pad

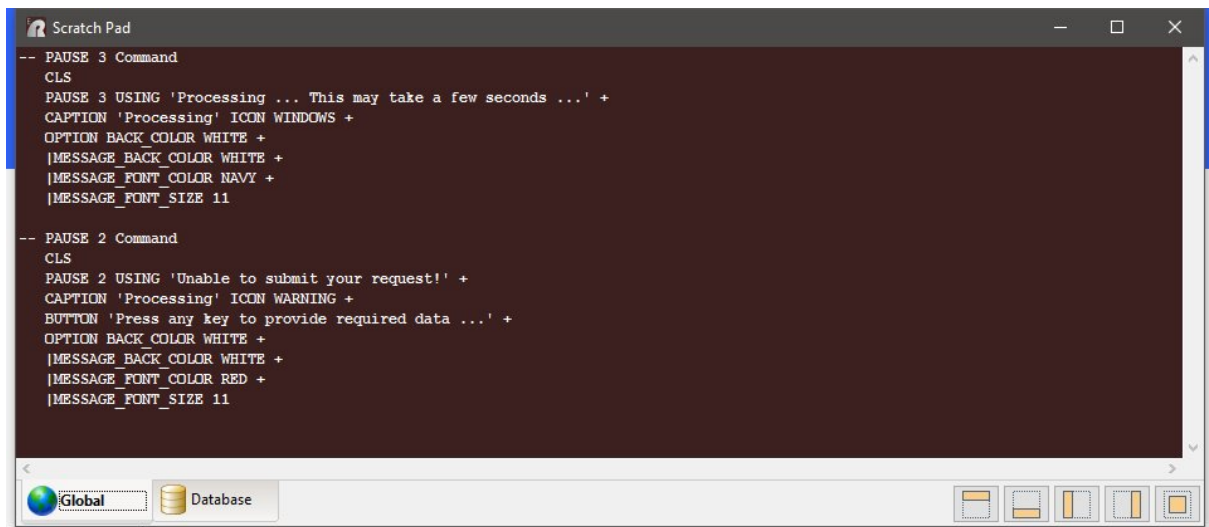
Did you know that you can keep track of all your development notes, and favorite command templates, etc., while developing applications?

These notes can be general (Global) or specific to the connected database.

A new "Scratch Pad" utility has been implemented to increase productivity during active development.

The scratch pad can hold any text or command template that is frequently used during development. The Scratch Pad utility window can be positioned several different ways to not obstruct R:BASE use. Global and Database tabs are available for database-specific or global development.

The Scratch Pad is available from the Utilities menu bar option, and can be displayed using the [Ctrl +F11] key combination, on demand.



Supported Versions:

- R:BASE X (Build...: 10.0.1.10909 or higher)

1.42 What Version of Engine DLL, and Files In Use?

Did you know that you can determine the exact version of the R:BASE Engine DLL, and all related Windows DLLs that are being used with your currently installed R:BASE?

Here's how to determine the specific version of Engine DLLs are being used with your R:BASE:

1. Start R:BASE
2. From the main window, select "Help" > "File Versions..." from the main Menu Bar.

Notice a list of all DLLs, and Plugins if loaded, along with details including Name, Version, Size, Company,

and Path.

You may select to Print the list to the following:

- Printer
- Text File
- PDF File
- Clipboard

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

Part



2 Settings

2.1 About the COLCHECK Setting

The additional COLCHECK setting has been introduced to manage common column names and corresponding data types when used as ALIAS names, while defining a new view or modifying an existing view.

COLCHECK specifies if columns will be checked for consistency when making views. When COLCHECK is ON (the default), view alias/column names will be checked for consistency to other occurrences of those names in the database, when the view is created.

Syntax:

```
SET COLCHECK ON/OFF
```

Similar checking is already in place when creating or modifying tables, and when attaching tables to a database, where R:BASE requires an alias that does not cause name/data type conflicts.

If an application requires COLCHECK be off to define a view, the proper results should not be expected once COLCHECK is switched back to ON when using the view. If a view needs to be created with COLCHECK OFF, it also needs to be used with COLCHECK OFF.

An impact may occur for a view that has already been defined, because when COLCHECK is ON, the values generated for the view will follow the data type of the alias. If an error occurs, the view definition should be reviewed.

2.2 Assigning Default Component IDs for Form, Label, and Report Controls

Did you know that while in the Form, Label or Report Designer, placing a new control can also assign a default Component ID to that particular control?

Of course, you can always overwrite the default name assigned, if you wish.

This enhancement is implemented to ameliorate the use of the PROPERTY and GETPROPERTY commands.

To incorporate this cool feature, you must first enable the check box for "Add Default Component ID" within your Form, Label, and Report Default Settings.

For the Form Designer, select "Settings" > "Form Designer" from the main Menu Bar. Within the "Form" tab, notice the "Add Default Component ID" option.

For the Report/Label Designers, select "Settings" > "Report/Label Designer" > "Default Settings" from the main Menu Bar. Within the "General" tab, notice the "Add Default Component ID" option.

2.3 AUTOMUMbering in a Multi-User Environment

In a typical multi-user, multi-session environment, you may notice the resource conflict or I/O -ERROR- when resources are not available, mainly from trying to get a lock on the SYS_DEFAULTS table to update the AUTONUM values.

Did you know that by increasing the setting for WAIT and INTERVAL in a multi-user, multi-session environment can reduce the instances of resource conflicts?

You can still get a resource not available message, but the chances of getting it can be reduced by increasing the value for WAIT from 4 (default) to something larger.

SET WAIT sets the minimum number of seconds to retry the last requested resource (a table or

database) before halting execution. Rather than aborting execution, SET WAIT allows you to set a length of time for R:BASE to keep trying to access a resource. A message is displayed showing the approximate percentage of wait time remaining.

If you do not run a SET WAIT command, R:BASE automatically retries the locked resource for approximately four seconds before issuing a retry message.

You can also adjust the INTERVAL in which R:BASE tries the command during the SET WAIT period. The SET INTERVAL command specifies the time to elapse before R:BASE retries the command that caused a conflict within the waiting period.

Recent update of R:BASE include enhancements related to the resource not available and I/O -ERROR- in a multi-user, multi-session environment that loops adding rows with Autonumbered columns.

Alternatively, in a multi-user, multi-session, and/or on-line R:BASE applications using a common database, you may also take advantage of (NEXT(tblname,autonumcol)) function.

2.4 Automating the Directory for SCRATCH Files

Did you know that you can automate the default directory for the R:BASE/OTERRO Scratch files without knowing the user's rights to their temporary directory?

SET SCRATCH sets the drive and directory location for temporary files created when sorting data.

SET SCRATCH ON - stores temporary sort files on the database drive and directory.

SET SCRATCH OFF - stores files on the current drive and directory.

SET SCRATCH <path> - provides the path to the location where temporary files are stored.

The latest builds of R:BASE and Oterro include the enhancement to simply type "TMP" for SCRATCH option in your configuration file.

```
SET SCRATCH TMP
```

Updating the SCRATCH setting to TMP in the appropriate configuration files, R:BASE/OTERRO will use the user's TMP environment settings for SCRATCH files on startup.

Using the latest builds of R:BASE/OTERRO, you need to either update the line for "SCRATCH TMP" to automatically define the windows temporary environment or use "SCRATCH C:\TEMP", for example, in R:BASE/OTERRO configuration files. If you define the "SCRATCH C:\TEMP", make sure that you actually have the C:\TEMP directory. We suggest that you keep all configuration (*.CFG) file in Windows or WinNT directory.

This option is also helpful when you establish a universal naming convention (UNC) network connection for R:BASE database as a System DSN defined with UNC, such as "\\FileServerName\SharedDirectoryName\dbname.rb1". This type of environment requires you to sure that you have also set the path for R:BASE and OTERRO temporary SCRATCH files.

Use any text editor or R:BASE Editor to edit the configuration file to read SCRATCH settings as follows:

```
SCRATCH TMP
```

This will help eliminate all issues related to the access rights, disk space and so on, when running R:BASE/OTERRO on enterprise servers and when accessing R:BASE database defined as a System DSN using the UNC option.

2.5 Cleaning Scratch (.\$\$\$) Files on Exit

Did you know that you can have R:BASE clean any remnants of temporary (.\$\$\$) scratch files, on exit? Here's how:

From the R:BASE main Menu Bar, select "Settings" > "Clean Scratch Files on Exit".

Checking this option will clean any remnants of temporary (.\$\$\$) scratch files, on exit.

2.6 Customizing Default Printer Settings

Did you know that among many default settings, now you can also set the default settings for your Report Printer, Paper Size and Paper Source?

Here's how:

1. Database Explorer > Settings > Report/Label Designer > Default Settings
2. Click on [Printer] tab and select the default Printer for report
3. Click on [Paper Size] tab to select the default Paper Size
4. Click on [Paper Source] tab to select the default Paper Source

These default settings are for your local workstation and will be effective for designing new reports only.

2.7 Customizing Hints

Displaying hints on your cool custom-designed forms, external forms, or even from the R:BASE Database Explorer can be very helpful for the end user.

Until now, you had the option to set background color, font, font style, font size, font effects, font color, and the time (in milliseconds) to display and hide the hint. The entire hint displays in a balloon.

What if your end-users don't like the hint embedded in the balloon? What about adding a balloon hint with an appropriate icon to display the nature of the hint?

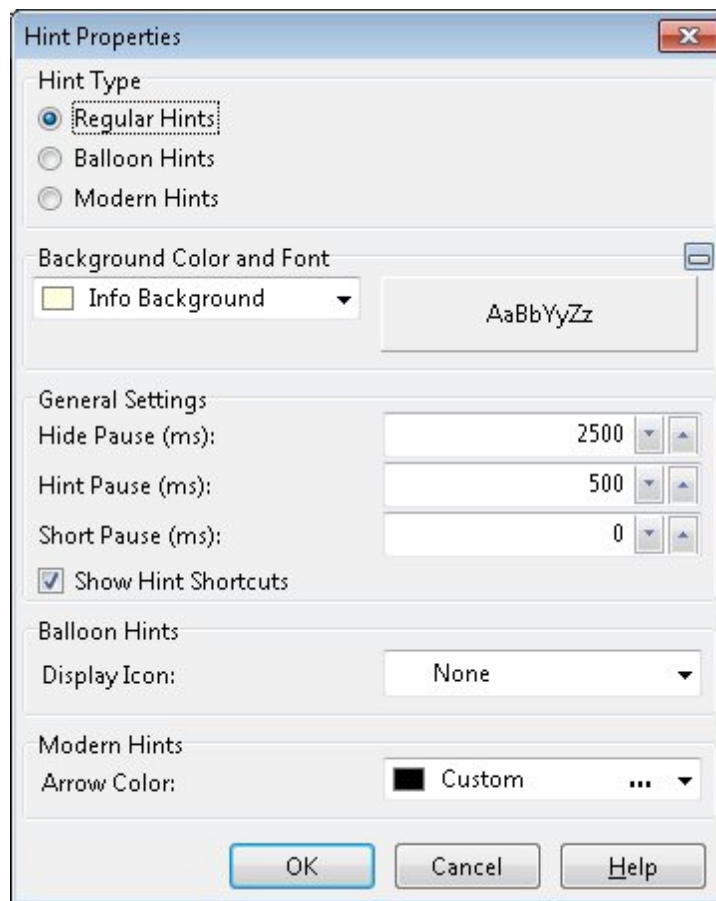
Did you know that you can now customize the hint settings to display hints with or without a balloon along with the option to use the appropriate icon when used with balloon?

Here's how:

1. Start R:BASE for Windows
2. Customize Hint Settings

Select "Settings" > "Hint Settings..." from the Menu Bar.

Notice the following options for Hint Settings:



⇒ **Hint Type**

Specifies the hint type

⇒ **Background Color and Font** Specifies the background color and text font, color and style. The Background Color only effects "Regular" and "Balloon" hints.

⇒ **General Settings**

Hide Pause (ms) - specifies the time interval for the hint to be displayed on the screen

Hint Pause (ms) - specifies the time interval for the hint to appear after the cursor first moves over the control

Short Pause (ms) - specifies the time interval for the hint to appear after the cursor leaves and immediately returns to the same control.

Show Hint Shortcuts - hint shortcuts will be displayed

⇒ **Balloon Hints**

Display Icon - specifies the icon that can be displayed within balloon hints

⇒ **Modern Hints**

Arrow Color - specifies the color of the hint arrow

3. If you wish to remove the balloon for hints, select the "Regular Hints" option for Hint Type.
4. If you wish to display the balloon hint with no icon, check the option for "Show Balloon" and select "None" from the list of available Display Icons
5. If you wish to display the balloon hint with an icon, check the option for "Show Balloon" and select one of the following icons from the list of available Display Icons.
6. Click the "OK" button to save the Hint Settings.

These hint settings will be saved for this workstation.

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

2.8 Customizing WALKMENU Time Interval Settings

Did you know that you can customize the WALKMENU Time Interval Settings to match your typing speed?

R:BASE now includes a Walkmenu Time Interval Setting to control the WALKMENU speed for CHOOSE, ListBox, ComboBox and Pop-up Menus.

1. Start R:BASE
2. Database Explorer > Main Menu > Settings ...
3. Select Walkmenu Time Interval ... and enter the value (default value is set to 1500 ms.)

The time interval value must be specified in milliseconds.

For example:

1000 for 1 second
1500 for 1.5 seconds

Using the command file or in an EEP, you may also control the WALKMENU time interval setting by using the PROPERTY command.

Example:

```
SET WALKMENU ON  
PROPERTY APPLICATION WALKMENUTIMEINTERVAL nnnn
```

About WALKMENU:

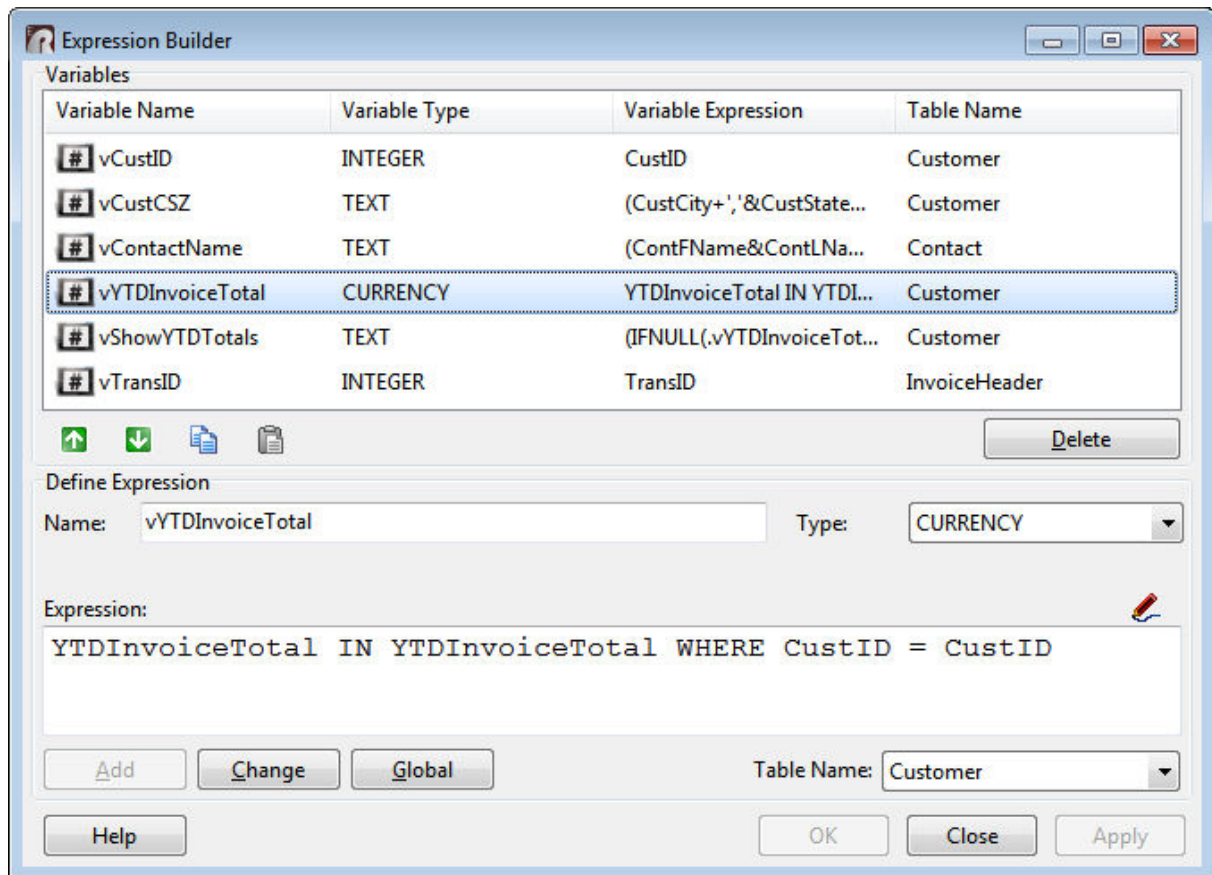
SET WALKMENU ON allows the user to access list item selections by typing the beginning characters (up to when a match is made) of their names.

2.9 Customizing the Forms/Reports/Label Expression Builder

While in Form, Report, or Label Designer, defining complex expressions and paying attention to commas, periods, quotes, parenthesis, etc., can be tedious to see when the default font size is set to Tahoma (Style: Regular, Size: 8, and Color: Black). There are times when users have to copy and paste the entire expression onto their favorite notepad, increase the font, view everything, correct missing periods, commas, or whatever, and then paste back into the expression builder.

Did you know that now you can customize the "Expression:" memo window to please your eyes?

The latest R:BASE eXtreme 9.5 includes an enhancement to customize and "Set Expression Memo Font".



Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

2.10 Database Stored Settings

Did you know that there are several R:BASE settings that are stored within the database files, which are automatically applied to any other R:BASE instance when it connects to that same database?

These are called Database Stored Settings. The following is a list of R:BASE Database Stored Settings:

CHARacter Settings:

BLANK
 DELIMIT
 LINEEND
 SEMI
 QUOTES
 PLUS
 SINGLE
 MANY
 IDQUOTES
 NULL

DATE Settings:

DATE FORMAT
 DATE SEQUENCE
 CENTURY Threshold

Default CENTURY

TIME Settings:

TIME FORMAT
TIME SEQUENCE

CURRENCY Settings:

CURRENCY SYMBOL
CURRENCY LOCATION
CURRENCY DIGITS
CURRENCY CONVENTION

Operating Conditions:

AUTOSKIP
BELL
CASE
NOCALC
REVERSE
TOLERANCE
ZERO

Database Events:

AFTER ERROR
BEFORE CONNECT
FIRST CONNECT
AFTER CONNECT
BEFORE PACK
AFTER PACK
BEFORE DISCONNECT *
AFTER DISCONNECT *
BEFORE EXIT *

* Specific to R:BASE X Enterprise

Supported Versions:

- R:BASE X (Build..: 10.0.1.10522 or higher)

2.11 Disabling TRACE in Forms, Labels and Reports

When you are close to distributing your developed R:BASE applications and are reviewing your work from a standpoint of securing your code, you may notice that the "SET TRACE ON" setting allows for the viewing of your command syntax within Form/External Form "On Before Start" EEPs and within Label/Report "Before Generate..." custom actions.

Even if the first command in that EEP is "SET TRACE OFF", the form, label or report won't trace after that command; however, all of the code can be seen by scrolling down before the F10 key is pressed.

Sure, you can password protect your form, but you don't want your users to have to enter a password every time they use the form just to keep the code safe from an unscrupulous person who happens to have R:BASE (even the trial version will trace!).

Did you know that you can disable the TRACE or ECHO by using the newly implemented option of "Disable Trace" while in Form, Label or Report Designers?

Now you can secure your code using the "Disable Trace" option. Here's how:

Forms/External Form File:

1. In Form or External Form Designer, select "Form Properties", and choose the "Miscellaneous" tab.
2. Notice the "Disable Trace" checkbox option.

Labels:

1. In the Label Designer, select "Label" from the main Menu Bar.
2. Notice the "Disable Trace" option.

Reports:

1. In the Report Designer, select "Report" from the main Menu Bar.
2. Notice the "Disable Trace" option.

2.12 Duplicating Your Personalized R:BASE Settings

When installing R:BASE on your computer, the default settings are automatically loaded. Then, you may further customize the R:BASE settings on your working desktop, including the look and feel, as well as all default settings when designing a new database, tables, forms, labels, reports, external form files, and applications as follows:

- Startup Options
- Database Explorer
- R:BASE Editor
- R> Prompt
- Data Browser
- Data Designer
- Form Designer
- Report/Label Designer
- Application Designer
- BLOB Editor
- Hint Settings
- Walkmenu Time Interval
- Default Printer Font
- Custom Colors
- Registry Settings
- Show Main Toolbar
- Warn When Closing R:BASE Session
- Show Check Box to Suppress Error Messages
- Clean Scratch Files on Exit
- Configuration Settings

These custom settings are saved and are effective when you are the same logged in user on that particular workstation. Any change in the layout afterward is also updated immediately, including Favorite Folders.

If you are in a situation with many workstations, such as office, home, two notebooks for the road, remote client(s) workstations, etc., you will have to either customize each individual workstation to match your favorite settings, custom colors, layouts, etc., one at a time, or live with what you have and customize as you go. This is not a productive R:BASE development environment, wouldn't you agree?

Did you know that you can export your R:BASE settings (not including the license) from one workstation to another by simply using the new "Registry Settings | Save to File" option?

Here's how:

01. Database Explorer | Main Menu | Settings ...
02. Select Registry Settings ...
03. Click on Save to File ...
04. Assign the file name and click on the [Save] button to save. The binary file will be saved as filename.bin

Keep this binary file handy when re-setting R:BASE settings on the same computer or duplicating your personalized settings on other computer(s). Use "Load From File ..." option to load settings.

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

2.13 Finding Default vs. Active SCRATCH Settings

The (CVAL('SCRATCH')) function returns the value of the currently active SCRATCH setting.

SET SCRATCH sets the drive and directory location for temporary (.\$\$\$) files created when managing TEMPORARY Table, TEMPORARY Views, TEMPORARY SATTACHed Tables, and Sorting data.

When using the SET SCRATCH ON, the temporary (.\$\$\$) files are stored in the database drive and directory.

When using the SET SCRATCH OFF, the temporary (.\$\$\$) files are stored in the current (startup) drive and directory.

When using the SET SCRATCH TMP, (default configuration), the temporary (.\$\$\$) files are stored in the Windows User Environment Variables folder set for TMP folder.

When using the manual SET SCRATCH <path>, the temporary (.\$\$\$) files are stored in the defined path folder location. When doing so, make sure that the folder does exist, and the user has Full Control and permission to such folder.

By default, R:BASE configuration files include the TMP for SCRATCH setting. This default will allow R:BASE or Oterro sessions to use the user's TMP environment settings for SCRATCH files on startup and eliminate all issues related to setting the SCRATCH directory and related files.

To take advantage of this setting, use the option "SCRATCH TMP" in the appropriate R:BASE and OTERRO configuration files. You may also use the "SET SCRATCH TMP" command in your application startup files.

Using the SET SCRATCH <ValidPathAndFolder> command at the R> prompt, in a command file or in application startup file, will overwrite the default SCRATCH setting defined in the R:BASE eXtreme 9.5 Configuration file.

Using the SET SCRATCH <path> command, and providing the invalid path and folder will result in an -ERROR- as follows:

-ERROR- Cannot create temporary file. (2964)

In that case, make sure that you provide the correct path and folder with full control and permissions.

In a scenario where you have overwritten the default SCRATCH setting, did you know that the R:BASE Settings will show you the "Default" and "Active" SCRATCH settings?

Here's how:

From the R:BASE main menu bar, select Settings > Configuration Settings

Within the Configuration Settings dialog, select the Scratch tab

Notice the illustration, including:

[x] TMP (to store temporary files in user environment TMP directory)

Default SCRATCH Location:

Active SCRATCH Location:

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

2.14 Handling Form/Report/Label Designer -ERROR-Messages

Did you know that you can pre-define all required global variables for Forms, Reports and Labels within the Form, Report or Label Designer?

To avoid any inconvenience the next time you open the form, report or label with -ERROR- messages related to missing global variable(s) in expression, a new "On Before Design Action..." has been implemented for Form, Report and Label Designers to pre-define all required variables with appropriate data types before opening the form, report or label in designer.

Here's how:

Form Designer | Layout | On Before Design Action...

Report Designer | Report | Actions | On Before Design...

Label Designer | Label | Actions | On Before Design...

2.15 New PAGELOCK Setting

Did you know that R:BASE X/X Enterprise (Version 10) includes a new (actually separate) PAGELOCK operating condition, which specifies how R:BASE locks data in a multi-user environment, based on the position of the row in the data file?

Sound familiar? It should!

Previously the R:BASE operating condition was controlled by QUALCOLS as a dual-functioning setting, specifying unique columns for ODBC tables and for page locking.

Now page locking (PAGELOCK) is separate from ODBC unique columns (QUALCOLS).

What does this mean for R:BASE X/X Enterprise users?

Application code must be updated to add PAGELOCK and set the condition to ON/OFF appropriately where QUALCOLS was used for page locking. Examples below demonstrate the setting could also be used dynamically in case-by-case instances to increase multi-user performance.

To summarize the code changes, if you used SET QUALCOLS 2 for faster page locking, add SET PAGELOCK OFF to your code. If you used SET QUALCOLS 10 to enforce a page lock for a faster UPDATE process, add SET PAGELOCK ON to your code.

PAGELOCK specifies how R:BASE locks data when updating and deleting rows.

The settings for PAGELOCK are:

ON - R:BASE uses page locking or row locking as appropriate. When PAGELOCK is ON and two or more users are updating rows within the same page of data, R:BASE only lets the first user update rows--the other users are locked out until the first user's update has been completed.

OFF - R:BASE uses a fast row-locking method where only row locking is applied with no page locking. When PAGELOCK is OFF, you can lock rows of data instead of locking a page of data. You increase multi-user performance when PAGELOCK is OFF. And even more so when STATICDB and FASTLOCK are on.

If you know that your application mainly updates or deletes data a row at a time, rather than many rows, set PAGELOCK to OFF for row locking. In this case, R:BASE locks a row, reads the row, makes the change, and then releases the row. Otherwise, set PAGELOCK ON for page locking when you are doing an UPDATE and/or DELETE affecting many rows in a table.

Keep in mind that the PAGELOCK setting can be changed dynamically and can be different for different users using the same database.

Technically, the most efficient and fastest method for updating data in multi-user environment is to SET

STATICDB ON, SET FASTLOCK ON, and SET PAGELOCK OFF. This particular combination will result in the fewest contentions between users.

Notes:

- FASTLOCK and PAGELOCK can be set on at the same time.
- Setting STATICDB and FASTLOCK to ON (in that order), with PAGELOCK set to OFF will significantly increase multi-user performance with individual row changes.
- PAGELOCK is not the same as SET ROWLOCKS.
- Setting the value of PAGELOCK does not change the setting of ROWLOCKS.
- The PAGELOCK setting can be changed dynamically and can be different for different users using the same database.

Example:

```
-- The UPDATE must alter the values for many rows
SET FEEDBACK ON
SET PAGELOCK ON -- use page locking
UPDATE <tablename> SET <columnname> = value -- no WHERE Clause
SET PAGELOCK OFF -- use row locking
SET FEEDBACK OFF
CLS
```

Supported Versions:

- R:BASE X (Build..: 10.0.1.10522 or higher)

2.16 Preserving Computed Column Values When Using UNLOAD/LOAD

Did you know that users can preserve original computed values when using UNLOAD or LOAD commands to move data, or when rebuilding the entire database with UNLOAD ALL?

A new database settings of NOCALC has been implemented to allow users to preserve original computed values when using UNLOAD/LOAD to move data, or when rebuilding a database with UNLOAD ALL.

Syntax:

```
SET NOCALC ON/OFF
```

Default: OFF

NOCALC suppresses or processes computed column expressions with the UNLOAD and LOAD commands.

If NOCALC is set to ON, then the UNLOAD command, as it creates LOAD statements, will generate new "load only" commands as it starts each table. These "load only" commands would be CALC and NOCALC, which operate in a fashion similar to CHECK/NOCHECK and FILL/NOFILL and NUM/NUM commands. The UNLOAD command will output the current values for the computed columns as it unloads the data.

In processing a LOAD command for a table, if it encounters a NOCALC command, then regardless of the current setting for NOCALC, the LOAD command will expect values for every column, whether computed or not. The values from the input would then be stored in those columns that are computed, rather than doing the actual computation.

Other methods for adding rows to a table, such as APPEND , the Data Browser, or a form, would all still calculate each computed column. Only the LOAD command with its special CALC/NOCALC commands could input a value into a computed column without doing the computation.

R:BASE stores the setting with the database.

Supported Versions:

- R:BASE X (Build..: 10.0.1.10522 or higher)

2.17 Turning Off User Interface Updates to Increase Performance

Did you know that you can now turn off Database Explorer update notifications in order to increase application performance?

A new operating condition setting, UINOTIF, is available to display or suppress table, view, stored procedure, form, report, and label user interface updates at the Database Explorer.

Syntax:

```
SET UINOTIF OFF - suppresses Database Explorer update notifications
SET UINOTIF ON (default) - displays Database Explorer update notifications
```

With UINOTIF set to ON, new objects (tables, views, reports, etc.) will be automatically refreshed and displayed within the user interface.

Setting UINOTIF to OFF will not automatically refresh and display new objects within the user interface. Any new objects still exist, they are simply not automatically displayed. Pressing [F5] will refresh the display and any new objects will appear.

The UINOTIF setting is located within the R:BASE configuration file, and is listed in the Configuration Settings dialog, within the "Display" tab.

The SHOW command can be used to display the current UINOTIF value:

```
SHOW UINOTIF
(UINOTIF ) ON Front-End User Interface Notifications
```

As the Database Explorer is not used in most instances of custom application use, this option offers a fast and easy boost for performance!

2.18 Using Conditional TRACE in a Command File

Did you know that you can use the conditional TRACE in a command file or EEP to just debug a block of code?

A new conditional SET TRACE ON/OFF option has been implemented to execute TRACE (Interactive Command File Debugger) inside a command file or EEP to trace a block of code as defined.

Syntax:

```
SET TRACE ON/OFF (Default is ON)
```

Example 01: (typical command line option)

```
TRACE filename.ext
```

Example 02: (in a command file)

```
-- start
Your code here ....

SET TRACE ON {this will start the trace within a command file}

Your code here ...

SET TRACE OFF {this will stop the trace within a command file}

Your remaining code here ...
```

-- end

Notes:

- Once you turn OFF the TRACE in an R:BASE session, you will need to turn it back ON.
- (CVAL('TRACE')) will return the current status of TRACE (Values: ON or OFF)
- SHOW TRACE will display the current status of TRACE (Values: ON or OFF)
- Newly created R:BASE configuration files (.CFG) will include the option for TRACE ON
- If you want no one to TRACE your code, setting the TRACE option to OFF at the beginning

2.19 Using Double-Click to Bring Up the Designers

Did you know that you can double-click on any Table, View, Stored Procedure, Report, Label, Form, External Form File, Application, or Command File to bring up the appropriate item in the corresponding designer?

After connecting to a the database and using the Database Explorer, traditionally, you would first highlight the appropriate Table/View/Stored Procedure/Report/Label/Form/External Form File/Application/Command File and then right click to access the speed menu or click on the "Design" option to open the selected item in its designer.

By default, double-click is used to "Open" or "Run" selected item. However, if you wish to simply double-click on the item to open it within its designer, all you need to do is customize the Database Explorer settings.

Here' how:

Database Explorer > Main Menu > Settings > Database Explorer ...
Click on "Double Click Open Designer" option to Select

These custom settings are specific to the logged in user only. Using these custom settings, you can run R:BASE Your Way!

2.20 Using the CHECKPROP Setting

In earlier R:BASE builds, any invalid or mistyped PROPERTY/GETPROPERTY "property" keywords were ignored and no validation was performed to identify the invalid "property" value used.

Did you know that PROPERTY and GETPROPERTY command errors can now be displayed if a "property" value is invalid? This enhancement applies to both forms and reports.

Syntax:

```
SET CHECKPROP ON/OFF
```

The default value is set to OFF.

The following is a simple example to illustrate the use of the setting. Let's say your form's On After Start EEP includes the following code:

```
PROPERTY <ComponentID> VISBLE 'FALSE'  
RETURN
```

Notice the misspelled text "VISBLE". The valid keyword is actually "VISIBLE". Before, this was ignored and the property keyword was not validated, without an ability to check the use of the command. New set CHECKPROP to ON.

```
SET CHECKPROP ON
```

Now if you run the form with same misspelled property keyword, you will get an -ERROR- Property 'VISBLE' not found in component <ComponentID> or the assigned value is invalid.

The setting will help you to revisit your code so that you can correct the syntax as well as use a valid property command.

Note:

The errors are only displayed if a new operating condition CHECKPROP is set ON.

The CVAL function can be used to capture the CHECKPROP status.

```
SET VAR vChkPropStatus TEXT = (CVAL('CHECKPROP'))
```

The SHOW command will display the CHECKPROP status.

```
SHOW CHECKPROP
(CHECKPROP) ON Display PROPERTY/GETPROPERTY ERROR messages during
processing
```

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

2.21 Warning When Changing Table Structure

Features within R:BASE for Windows include a warning to users when accessing and managing the table structures (RBDEFINE) in a multi-user connected database environment, using the Database Explorer.

Did you know that now you can turn off that -Warning- message, if you are a veteran and know what you are doing, taking full responsibility for accessing or managing table structure in a multi-user environment?

Here's how:

01. Start R:BASE
02. Database Explorer | Main Menu | Settings | Data Designer ...
Notice the option for "Warn When in Multi-User Environment"
03. Un-Check this option to turn off the warning message.

The default installation is always set to checked.

Now you can have your cake and eat it too.

2.22 Warning When Closing an R:BASE Session

Have you ever been in a situation with many opened R:BASE sessions on your normal or extended desktop (because you can), and then mistakenly clicked on the close [x] icon of the R:BASE Database Explorer, closing the session (as it should), and you wished it would have warned you if you really wanted to close that session of R:BASE?

Now R:BASE includes a "Setting..." to prompt you if you really want to close an R:BASE session when clicking on the [x] close window icon of the Main Database Explorer.

Here's how:

1. Start R:BASE
2. Database Explorer | Main Menu | Settings

Check the Setting for:

Warn When Closing R:BASE Session (the default is set to Off - unchecked)

3. Now close the R:BASE session by clicking on the [x] and close window icon on the Main Database Explorer window.

Notice the confirmation dialog box?

"Do you really want to close this R:BASE session? [Yes] [No]"

Part



3 R> Prompt

3.1 Copying/Pasting Multiple Line Code

When copying your complete code with multiple lines and continuation symbols such as "+" at the R> prompt, it can be tedious to take out all the plus characters and turn the entire code as one line for the R> prompt to execute properly.

For example, a command looks like the following when typed in REdit or the R:BASE Editor.

```
CHOOSE vWhereCluase FROM #WHERE IN Customer +
  OPTION THEMENAME R:BASE Rocks! +
  |WHERE_CLAUSE WHERE CustState IS NOT NULL +
  GROUP BY CustState HAVING COUNT(CustState) > 1
```

If you copy and past the above command, with multiple lines and continuation characters at the R> prompt, it will be copied exactly as is. Then, you would have to eliminate the "+" characters before pressing the [Enter] key to execute the command.

Did you know that you use the [Shift]+[Ctrl]+[Alt]+[C] keys to copy the entire block, then paste it at the R> prompt as one single line without the continuation "+" characters?

Using the [Shift]+[Ctrl]+[Alt]+[C] keys eliminates the continuation "+" characters at the end of each line when pasted.

3.2 Documenting and Recycling the Command History

The enhanced Command History panel is a new addition within the R> Prompt window. If it is not displayed, you can access the Command History panel, using the [Ctrl]+[H] keys, or from within the R> Prompt Settings.

To execute any command displayed within the Command History panel, you can either double click on a command, or select the command from the list, which will display the command at the R> Prompt Input console, and then press the [Enter] key.

Did you know that you can [Save] the entire command history to a file or copy to the clipboard for documenting or recycling purposes? You can also [Load] the command history from file.

Notice the following options when you right-click on "Command History" panel.

- Run Selected Command
- Clear Command History
- Delete Selected Command
- Close Command History Window
- Add To Favorite Commands
- Run Favorite Command
- Load Command History from File...
- Save Command History to File...
- Copy Selected Command to Clipboard
- Copy Command History to Clipboard

Supported Versions:

- R:BASE X (Build..: 10.0.1.10522 or higher)

3.3 Hot Keys

Usually, when you type a command at the "Input Console" of the R> prompt and press the [Enter] key, the "Output Console" displays the output results. The typed command at the input console is then cleared.

If you wish to retype the same command again, you either have to use the [Page Up], [Page Down] or [Ctrl+H] keys to view the history of all commands typed at the R> prompt. The history is retained for the current session of R:BASE. Clicking the mouse on the Output Console and then pressing the [Page Up] key reveals the entire history of output results. Using the CLS command clears the buffer of all stored results in Output Console.

Did you know that you can type the command at the input console of the R> prompt and still retain the typed command by simply holding the [Shift] key and then pressing the [Return] Key?

Holding the [Shift] key and then pressing the [Return] key will retain the typed command in the input console of the R> prompt.

Part

IV

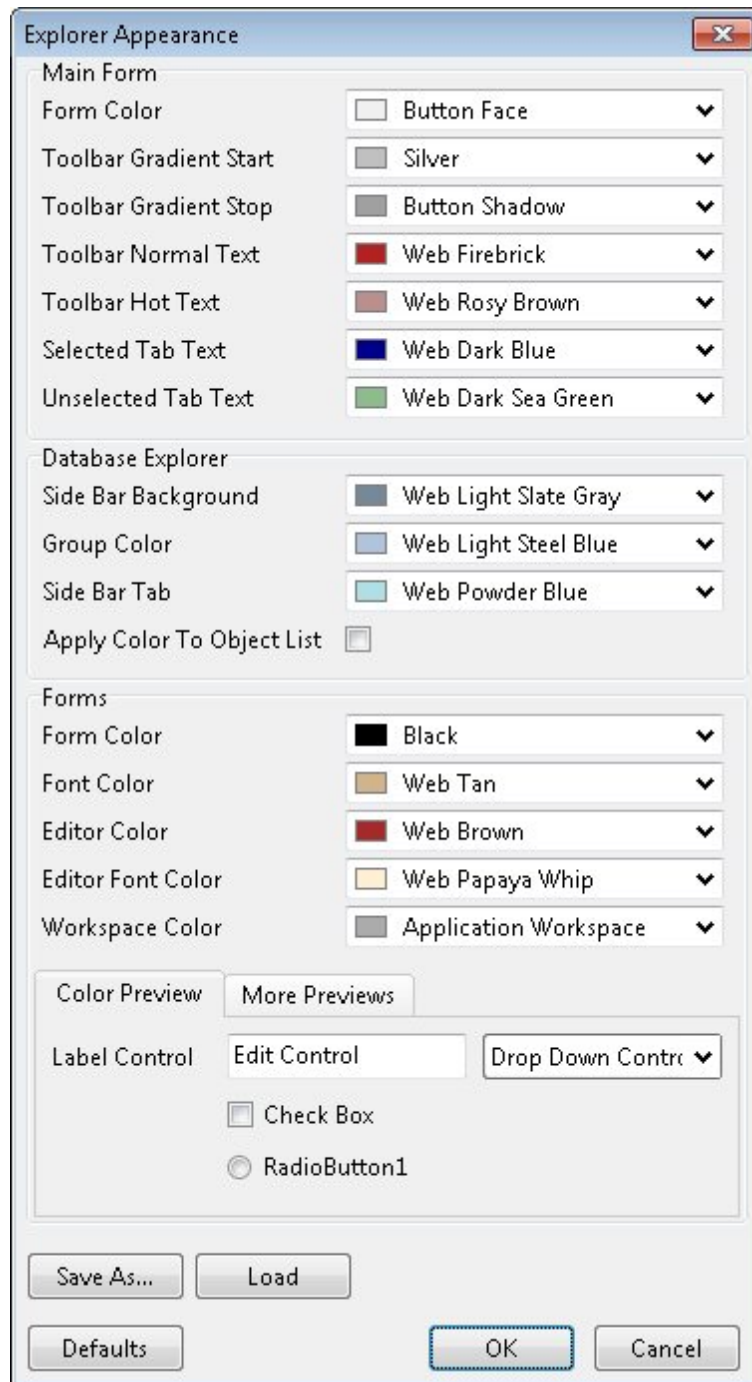
4 Database Explorer

4.1 Customizing the Database Explorer

Did you know that you can customize the appearance of Database Explorer, exactly the way you like it?

New Explorer Appearance settings are available to adjust the color schema of R:BASE and the Database Explorer areas.

- Settings > Database Explorer > Explorer Appearance



Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

4.2 Default File Names for Unloading Forms/Labels/Reports

Did you know that when you unload a form, label or report using the Database Explorer, the default file name includes a suffix "_Form", "_Label", or "_Report" for the form, label, or report?

Using this method, you will never have the problem of forms, labels, or reports with same names wiping out each other's LOB files.

For example:

When unloading a form named "Customers", the following two files will be created:

- Customers_Form.frm
- Customers_Form.lob

When unloading a label named "Customers", the following two files will be created:

- Customers_Label.lbl
- Customers_Label.lob

When unloading a report named "Customers", the following two files will be created:

- Customers_Report.rpt
- Customers_Report.lob

4.3 Exploring the Database Explorer

Here are a few search tips related to the Enhanced Database Explorer:

01. Renaming List Items

- Databases
- Tables
- Views
- Stored Procedures
- Forms
- Reports
- Labels
- External Forms
- Applications
- Command Files

Highlight the item, right-click and select "Rename ...".

You may also use [F2] key to rename selected item, or use click-pause-click to rename the selected item as well.

02. Stored Procedures - Find in Stored Procedures

From the Database Explorer, select Stored Procedures from the Group Bar menu on the left.

- I. Use [Ctrl+F] to launch "Find in Stored Procedures" Window to Find Text.
- II. Double-clicking on the selected item in Search Results will launch R:BASE Editor with highlighted found text. Use [F8] key to search next match while in the same block of code.

03. Forms - Find in Custom EEPs

From the Database Explorer, select Forms from the Group Bar menu on the left.

- I. Select a single, multiple, or [Ctrl+A] for all forms.
- II. Use [Ctrl+F] to launch "Find in Custom EEPs" dialog box to enter text to find.
- III. Selecting the form in [Form List] will expand Search Results with itemized list of controls and embedded Custom EEP.
- IV. Double-clicking on the link will launch R:BASE Editor with highlighted found text. Use [F8] key to search next match while in the same block of code.

Note:

While editing the Custom EEP, notice the Editor Caption with all details related to the Form Name, Control Name, Caption, Coordinates, and option that is used for Custom EEP, such as On Click EEP, etc.

04. Reports - Find in Custom EEPs

From the Database Explorer, select Reports from the Group Bar menu on the left.

- I. Select a single, multiple, or [Ctrl+A] for all reports.
- II. Use [Ctrl+F] to launch "Find in Custom EEPs" dialog box to enter text to find.
- III. Selecting the report in [Report List] will expand Search Results with itemized list of sections/bands and embedded Custom EEP.
- IV. Double-clicking on the link will launch R:BASE Editor with highlighted found text. Use [F8] key to search next match while in the same block of code.

Note:

While editing the Custom EEP, notice the Editor Caption with all details related to Report Name, Action, Band, and option that is used for Custom EEP, such as On Before Generate EEP.

05. Labels - Find in Custom EEPs

From the Database Explorer, select Labels from the Group Bar menu on the left.

- I. Select a single, multiple, or [Ctrl+A] for all labels.
- II. Use [Ctrl+F] to launch "Find in Custom EEPs" dialog box to enter text to find.
- III. Selecting the label in [Label List] will expand Search Results with itemized list of sections/bands and embedded Custom EEP.
- IV. Double-clicking on the link will launch R:BASE Editor with highlighted found text. Use [F8] key to search next match while in the same block of code.

Note:

While editing the Custom EEP, notice the Editor Caption with all details related to Report Name, Action, Band, and option that is used for Custom EEP, such as On Before Generate EEP.

06. External Forms - Find in Custom EEPs

From the Database Explorer, select External Forms from the Group Bar menu on the left.

- I. Select a single, multiple, or [Ctrl+A] for all external forms.
- II. Use [Ctrl+F] to launch "Find in Custom EEPs" dialog box to enter text to find.
- III. Selecting the form in [Form List] will expand Search Results with itemized list of controls and embedded Custom EEP.
- IV. Double-clicking on the link will launch R:BASE Editor with highlighted found text. Use [F8] key to search next match while in the same block of code.

Note:

While editing the Custom EEP, notice the Editor Caption with all details related to Form Name, Control Name, Caption, Coordinates, and option that is used for Custom EEP, such as On Click EEP, etc.

07. Applications - Find in Custom EEPs

From the Database Explorer, select Applications from the Group Bar menu on the left.

- I. Select a single, multiple, or [Ctrl+A] for all applications.
- II. Use [Ctrl+F] to launch "Find in Custom EEPs" dialog box to enter text to find.
- III. Selecting the application in [Application List] will expand Search Results with itemized list of controls and embedded Custom EEP.
- IV. Double-clicking on the link will launch R:BASE Editor with highlighted found text. Use [F8] key to search next match while in the same block of code.

Note:

While editing the Custom EEP, notice the Editor Caption with all details related to Application Name, Control Name, Caption, Coordinates, and option that is used for Custom EEP, such as On Click EEP, etc.

08. Command Files - Find in Files

From the Database Explorer, select Command Files from the Group Bar menu on the left.

- I. Select a single, multiple, or [Ctrl+A] for all command files.
- II. Use [Ctrl+F] to launch "Find in Files" dialog box to enter text to find.
- III. You may also customize the search using Search Options and File Attributes. Click on [Start] button to search.
- IV. Highlight one, multiple, or [Ctrl+A] for all command files that match the criteria. Click on [Open File(s)] button to open all command files in R:BASE Editor.

09. Command Files - Editing Multiple Files at Once

From the Database Explorer, select Command Files from the Group Bar menu on the left.

Select a single, multiple, or [Ctrl+A] for all command files and then right-click to select Edit File(s) to edit all files at once.

Make sure to customize R:BASE Editor Settings to take advantage of new and enhanced options, such as:

General (Background, Selected Block, Text Match)

- o String
- o Display
- o Line Highlight
- o Background
- o Printer Font
- o Compare by Content
- o Structure Toolbar
- o Syntax Schema

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

4.4 Form Preview From the Database Explorer

Did you know that you can preview any Form from the Database Explorer?

Here's how:

A new setting has been introduced to preview your form canvas from the Database Explorer, when you are not even using the form designer.

Steps to activate this "Preview" option:

1. From the Database Explorer, select "Settings" > "Form Designer..." from the Menu Bar.

The "Default Form Settings" dialog window will appear.

2. Check the option for [] Save Form with Preview
3. Click on [OK] button to save the user settings.

Now open any form in form designer, make any changes, save the form and then close the Form Designer. From now on, creating a new form or updating an existing form will be saved with the preview capabilities.

When using the [Forms] section of the Database Explorer, select that form you have just saved and then right-click on it for all speed menu options. Notice the additional item [Form Preview]. Selecting this option will display the Form Preview.

4.5 Key Word Search Options

Did you know that there are four (4) options to search for a key word?

1. [Ctrl] + F

- Find in Custom EEPs
 - Stored Procedures, Forms, Reports, Labels, External Forms,
 - Applications, Command Files
2. [Ctrl] + L
- Find in Control Properties - Captions
 - Forms, Reports, Labels, External Forms
3. [Ctrl] + [Shift] + F
- Find in Expressions
 - Forms, Reports, Labels
4. [Ctrl] + N (Search key words in Name or Comments)
- Tables
 - Views
 - Stored Procedures
 - Forms
 - Reports
 - Labels
 - External Forms
 - Applications
 - Command Files

Supported Versions:

- R:BASE X (Build..: 10.0.1.10522 or higher)

4.6 One Click UNLOAD for Data and Structure

Did you know that you can UNLOAD the STRUCTURE as well as the DATA (ALL), using the one-click option of the R:BASE Database Explorer?

Here's how:

```
CONNECT RRBYW19
```

Database Explorer | Tables

Right-click on the desired table and click on "Unload" > "Structure and Data..." option of the speed menu.

This will UNLOAD the data as well as the structure of selected table to an external file, using the default file name as tablename.all, which you can change, if you wish.

The other options for "Unload" include strictly structure output:

- Structure...
- Definition...
- Definition with Constraints, Indexes, Comments...

4.7 Open Table/View Options Using the Data Browser/Editor

Traditionally, when using the Database Explorer, double-clicking on a table/view will bring up the Data Browser in a Browse mode and then you can sort the data by clicking on a column header or using the Sort options from the main menu. You can also to press [F4] key or click on the toolbar icon to toggle Edit or Browse mode. There is no option to limit the number of rows with a WHERE clause after you have double clicked on the table/view.

Did you know that you can use the new Setting for the DB Explorer to display a "Open Table" dialog with the following options:

- WHERE Clause
- Open As: [] Browse [] Edit
- WHERE Builder
- Open As MDI Window

This setting will provide you with options to customize the data and Data Browser/Editor Window itself.

R:BASE Main Window | Main Menu | Settings | Database Explorer ...

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

4.8 Renaming Database Explorer Objects

Using the new Main Window and Database Explorer, while seeing the list of Tables, Views, Stored Procedures, Forms, Reports, Labels, External Form Files, Applications, and Command Files, now you can simply left click twice (not double-click) or use the [F2] key to rename the displayed name accordingly.

Be careful using this shortcut!

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

4.9 Resetting Version Number of External Form Files

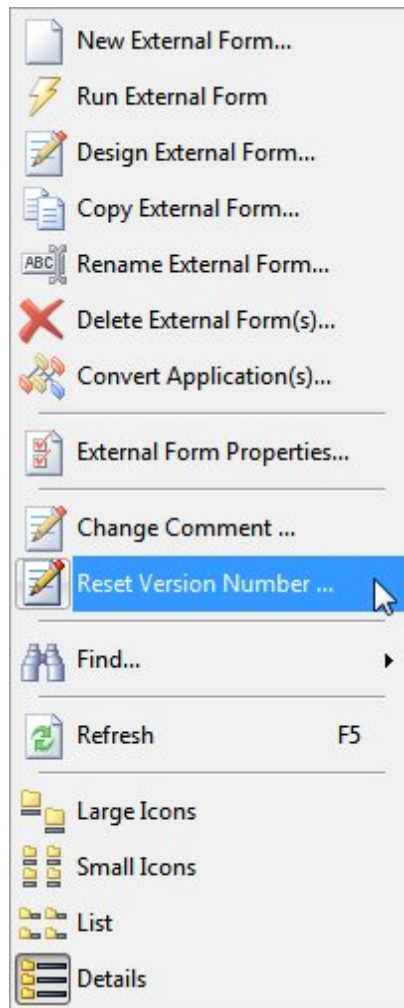
Similar to internal database forms, the External Form Files also include the "Version" control option to keep track of external form file revisions.

Did you know that you can reset the version number of External Form Files?

Here's how:

Within the Database Explorer, select External Forms ...

Highlight the External Form, Right-Click, and select Reset Version Number ...



In addition to resetting the version number, you may also change the comment for external form file, if you wish.

Database Explorer | External Forms ...

Highlight the External Form, Right-Click, and select Change Comment ...

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

4.10 Searching for Text in Control Properties

Did you know that you can search for specific text that may be defined in Form, Report, Label, and External Form control properties?

The new "Find in Control Properties" feature opens a search utility to locate specific text defined within the properties of any object for the connected database. The functionality to search for text within Control Properties allows users to locate Component IDs, hints, text captions, columns listed in WHERE Clauses for Lookup controls, and also pop-up menus. Any text defined within a form, report, or label control will be used in the search!

When searching for text within Control Properties, text strings can be located within the following modules:

- Forms
- Reports
- Labels
- External Forms

Here's how:

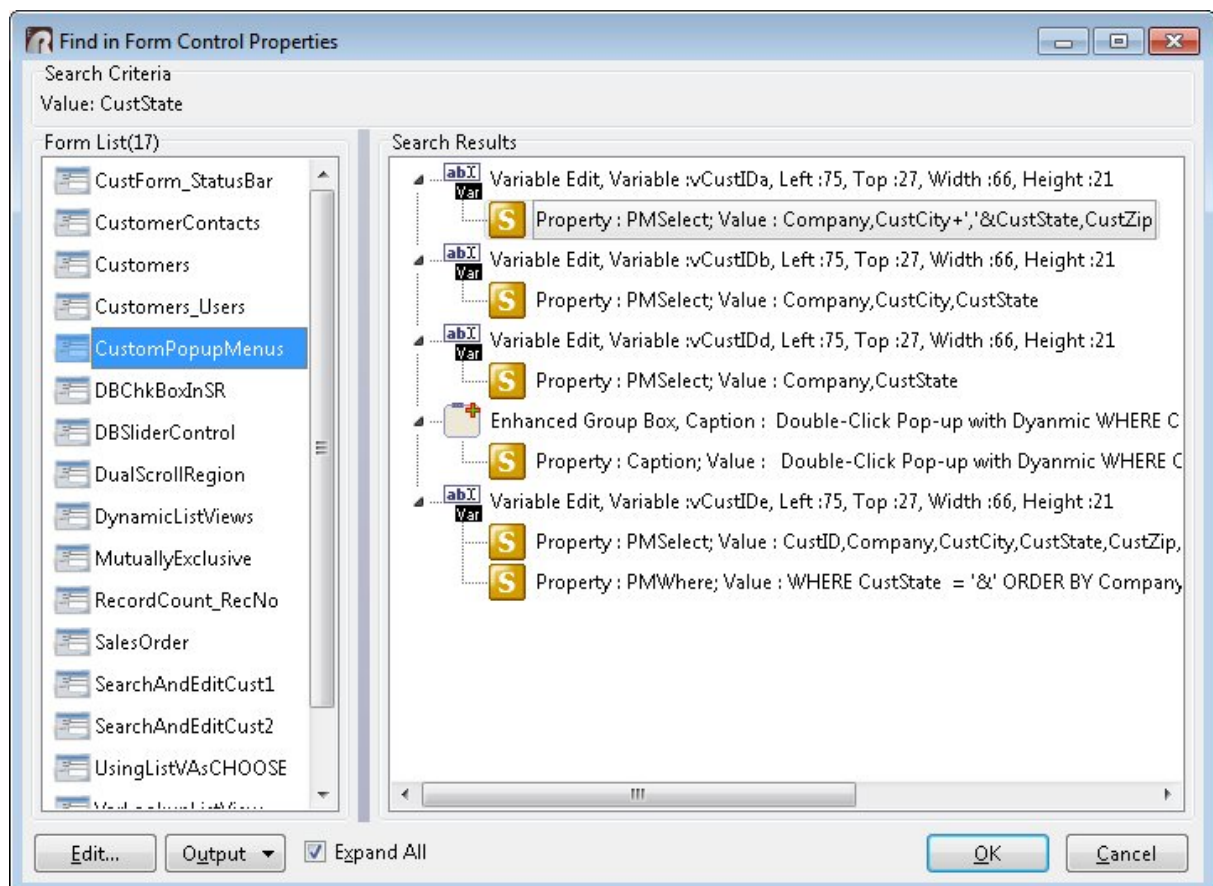
To begin searching, press the [Ctrl]+[L] key combination to display the "Find in Control Properties" dialog.



Searching for text can include all listed items displayed at the Database Explorer, or just a selected few. For example, selecting ten out of fifty forms, and then pressing [Ctrl]+[L] will search for the text for all controls within the ten selected forms.

Note: The "Find in Control Properties" dialog can only be launched from the Database Explorer.

After the search is complete, a list of forms and form objects that match the search will be displayed. The search results will contain form object(s) whose property and value match the search criteria.



The property value can be launched within a dialog window for editing by double clicking on it, or by selecting the "Edit..." button. The search results can be saved to several output formats.

The "Expand All" check box allows the list of EEPs in the search results to expand for full viewing.

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

4.11 Searching for Text in Expressions

Did you know that you can search for specific text that may be defined in Form, Report, and Label expressions?

The "Find in Expressions" feature opens a search utility to locate specific text defined within the form, report, and label expressions for the connected database. The functionality to search for text within expressions allows users to locate specific columns, tables, functions, variables, and any part of a WHERE Clause.

Any text defined within an expression will be used in the search!

Here's how:

When searching for text within expressions, text strings can be located within the following modules:

- Forms
- Reports
- Labels

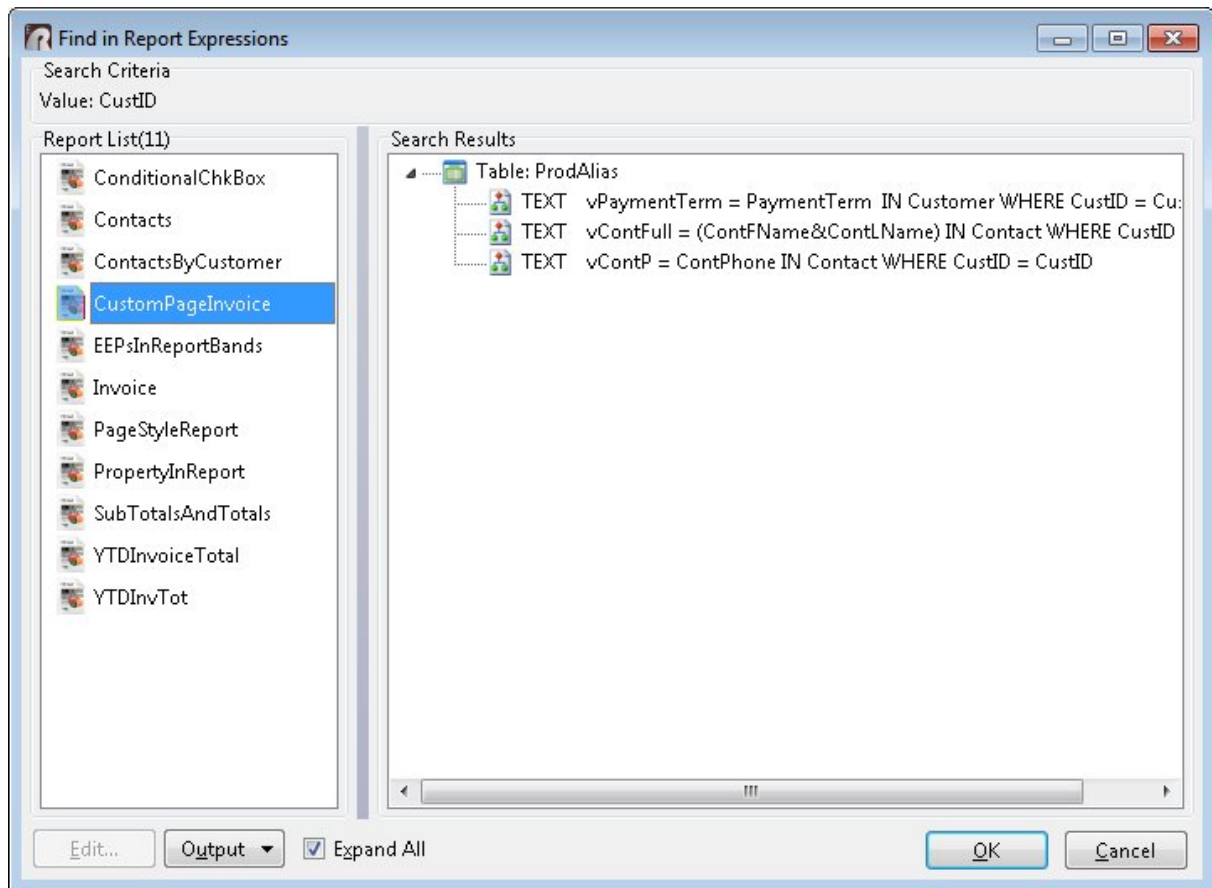
To begin searching, press the [Ctrl]+[Shift]+[F] key combination to display the "Find in Expressions" dialog.



Searching for text can include all listed items displayed at the Database Explorer, or just a selected few. For example, selecting fifteen out of fifty reports, and then pressing [Ctrl]+[Shift]+[F] will search for the text for all expressions within the fifteen selected reports.

Note: The "Find in Expressions" dialog can only be launched from the Database Explorer.

After the search is complete, a list of reports and defined variable expressions that match the search will be displayed.



The search results will contain the table(s) and expressions that match the search criteria. The variable expression can be launched within a dialog window for editing by double clicking on it, or by selecting the "Edit..." button. The search results can be saved to several output formats. The "Expand All" check box allows the list of expressions in the search results to expand for full viewing.

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

4.12 Searching in Database Explorer

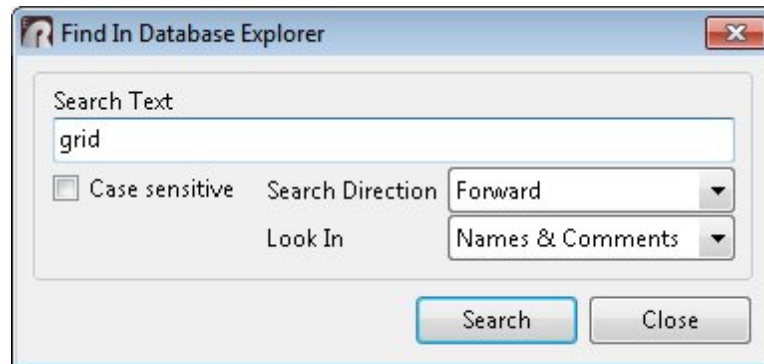
After CONNECTing the database, the Database Explorer includes the following sections and appropriate columns when configured as "View as Details":

- Tables (Name, Comment, Columns, Rows)
- Views (Name, Comment, Columns)
- Stored Procedures (Name, Comment, Size, Date/Time, Version)
- Forms (Name, Main Table, Comment, Size, Date/Time*, Version*)
- Reports (Name, Main Table, Comment, Size, Date/Time*, Version*)
- Labels (Name, Main Table, Comment, Size, Date/Time*, Version*)
- External Forms (Name, Comment, Size, Date/Time, Version)
- Applications (Name, Date/Time, Version)
- Command Files (Name, Extension, Date/Time, Size)

* = R:BASE X Enterprise only.

Did you know that you can use the "Ctrl + N" hot key to search for any key word in appropriate column while selecting the appropriate section of Database Explorer?

In a typical development environment, for example -- one remembers the form has to be something "... MAP ...". Using the [Ctrl] + [N] hot key to search text, all forms containing map in the name and comment will be highlighted, one at a time, for easy selection.



This feature is extremely helpful for those that have databases with significant number of tables, views, stored procedures, forms, reports, labels, external forms, applications, and command files.

Note: The [Ctrl] + [N] will not do a search in the active or selected row.

Supported Versions:

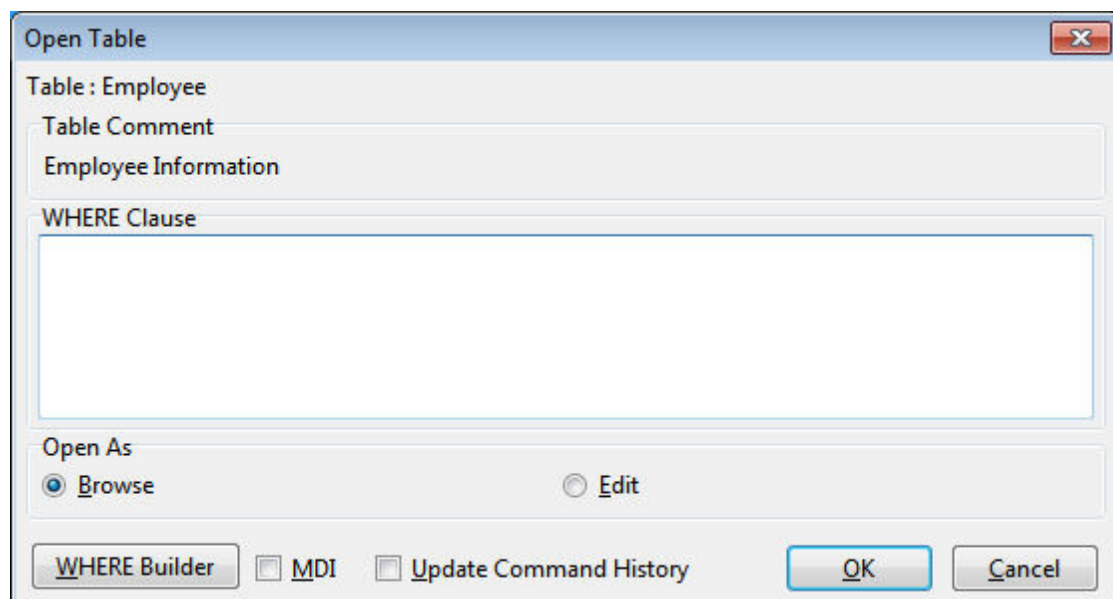
- R:BASE X (Build..: 10.0.1.10522 or higher)

4.13 Using MDI Option for Database Explorer Modules

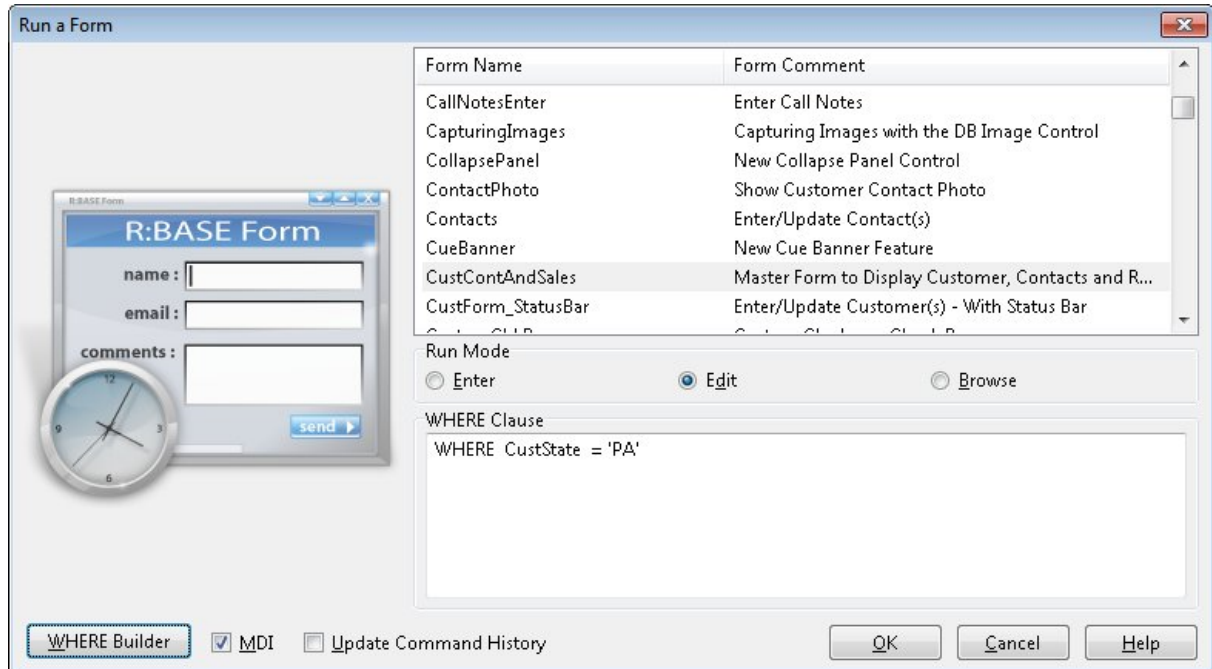
When customizing the settings for Database Explorer, in addition to change the appearance of the Database Explorer window, you may also select the option to "Use Dialog to Open Table/View".

Selecting this option toggles the ability to launch a dialog to add a WHERE Clause, change the browse mode, and use an MDI window when opening a table or view.

When you double click on a table or view, "Open Table" or "Open View" dialog box is launched and next to the WHERE Builder button an "MDI" check box is available. When MDI option is checked, the Data Browser will be displayed as an MDI windows. You may open as many tables or views at the same time using the MDI option.

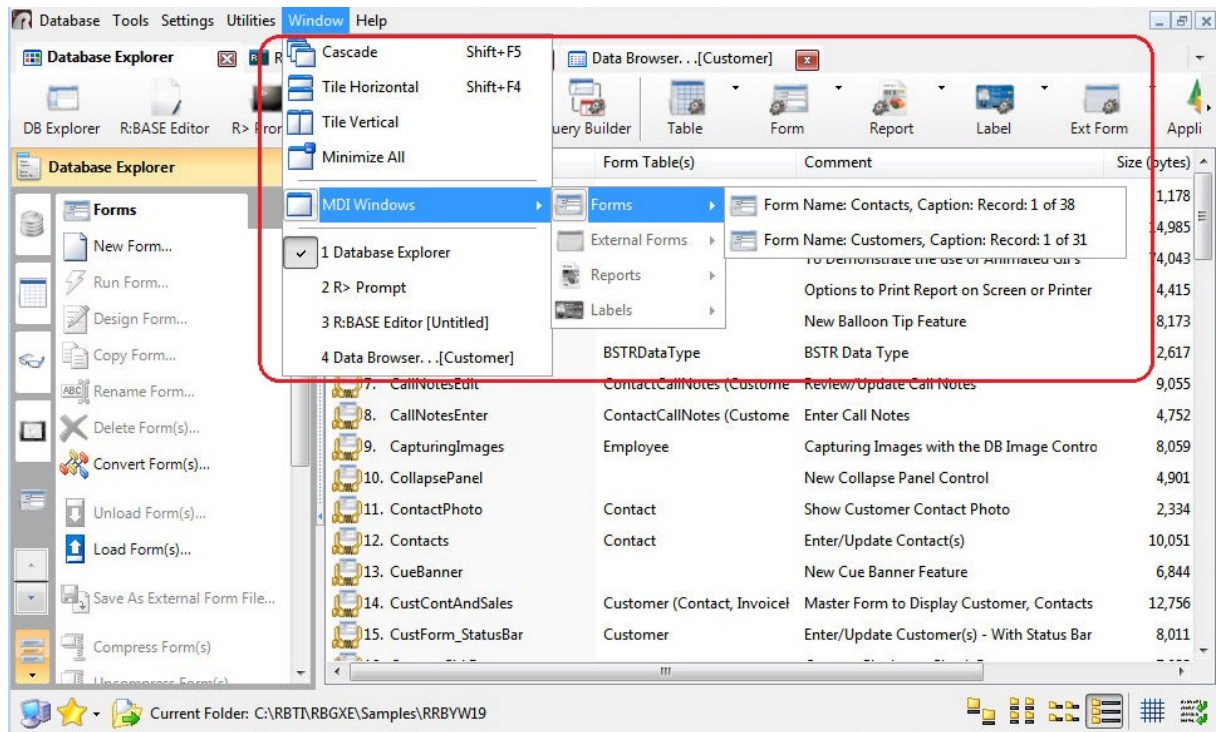


Did you know that you can use the MDI option when double-clicking on Forms, Reports, Labels, or External Forms to run? Notice the [] MDI option next to the [WHERE Builder] button.



You may also use the same option for Reports, Labels, and External Forms. There is no limit.

If you wish to retrieve, switch back to previously opened MDI Browser, Form, Report, Label, External Form, or would like to see a list of all opened MDI windows, you may use the Window option of the Database Explorer by selecting Window > MDI Windows.



Supported Versions:

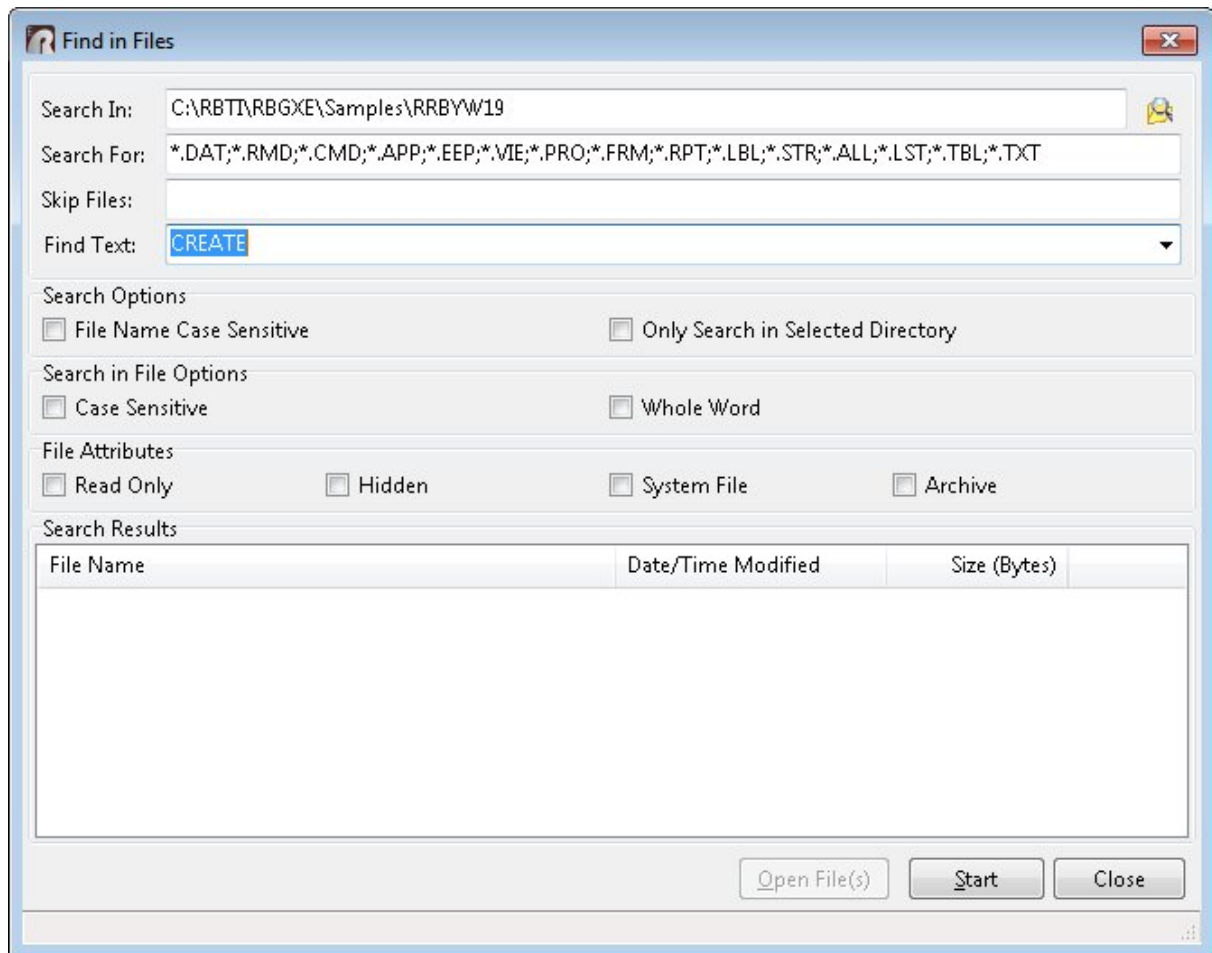
- R:BASE X (Build..: 10.0.1.10522 or higher)

4.14 Using the Find in Files Utility

The "Find in Files" utility allows users to locate command files based upon specific text within the files.

Two new "Search" options allow the search for text to be more precise. For example, with "Case Sensitive" and "Whole Word" enabled, a search for the command "CONNECT" will no longer return the unwanted results for DISCONNECT or SCONNECT, or any locations where the word "Connect" with lower case letters is used in a DIALOG or PAUSE as the message or caption.

From the Database Explorer, select "Command Files" > "Find in Files"



Search criteria can be modified based on file extensions, case sensitivity, sub-directory searches, and file attributes. After the search is complete, the selected files can be opened in the R:BASE Editor.

Supported Versions:

- R:BASE X (Build..: 10.0.1.10522 or higher)

Part



5 Data Designer (RDEFINE)

5.1 Computed Columns and Default Values

While upgrading and migrating legacy R:BASE databases to R:BASE X/R:BASE X Enterprise, a warning message will be displayed if the legacy table column contains a default value for a computed column.

If the command syntax to CREATE a table defines a default value on a computed column, the table will be created, however, a warning message will be displayed, but the default value on the computed column will not be added.

IMPORTANT NOTE:

This may alert some users if they inadvertently added a default value to a column definition, and then decided to change that column to a computed value, but did not remove the default value.

The R:BASE X/R:BASE X Enterprise Data Designer and the CREATE TABLE command has been enhanced to display warning message and ignore default value, if defined, for a computed column.

While converting legacy database(s), you will also notice the warning message when building a fresh database from an output file that was created using the UNLOAD command based on a legacy database, only if the build process finds a default value for computed column.

5.2 Reorganizing Columns in Data Designer

Did you know that you can reorganize/re-sequence table columns in the Data Designer using the Change Order [Move Up] and [Move Down] speed buttons?

Here's how:

1. Start R:BASE.
2. Connect to a database.
3. From the Database Explorer, select the "Tables" option.
4. Select the appropriate table and choose the "Design Table" option.
5. Within the "Data Designer", select the "Columns" tab.

Notice the green arrow [Move Up] and [Move Down] speed buttons!

6. Highlight the exact column that you wish to move up/down.
7. Click on the [Move Up] or [Move Down] button accordingly.

Notice the new position of selected column

8. Click on the "Save" button to save the current structure or "Save and Close" to save the current structure and close the data designer.

5.3 Validate the syntax for an email address

Here is a tip from the Father of R:BASE, Wayne Erickson (Head of the R:BASE Development Team), that validates the syntax for an email address in R:BASE.

Wayne's Corner: Validating syntax for an email address in R:BASE ...

For an example, assume you have a table called "person" and it has a column called "email" for storing email addresses. You want to check some things to make sure the values in this column meets some basic things an email address should have. Here is the rule definition code:

```
RULES 'Improper Email Address' FOR Person SUCCEED +  
WHERE +  
(SLOC(SMOVE(`Email`,`@`),SLOC(`Email`,`@`)+1),SLEN(`Email`)-
```

```
(SLOC(`Email`,`@`)),' ',1),'.')) > 1 +
AND +
((SLEN(SMOVE(`Email`,(SLOC(`Email`,`@`)+1),(SLEN(`Email`)-
SLOC(`Email`,`@`)),' ',1)))- +
(SLOC(SMOVE(`Email`,(SLOC(`Email`,`@`)+1),(SLEN(`Email`)-
SLOC(`Email`,`@`)),' ',1),'.')) >= 2 +
AND +
(SLOC(STRIM(LJS(`Email`,40)),`@`)) > 2
```

This particular rule checks three things:

1. Are there at least 2 characters before an "@" character? This is done with the

```
(SLOC(STRIM(LJS(`Email`,40)),`@`)) > 2
```

part of the rule.

2. Is there a "." after the "@", and is there at least one character between the "@" and the "."? This is done with the

```
(SLOC(SMOVE(`Email`,(SLOC(`Email`,`@`)+1),SLEN(`Email`)-
(SLOC(`Email`,`@`)),' ',1),'.')) > 1
```

part of the rule.

3. Are there at least 2 characters after the "." which follows the "@" character? This would allow something like x@x.ca but not x@x.c. This is done with the

```
((SLEN(SMOVE(`Email`,(SLOC(`Email`,`@`)+1),(SLEN(`Email`)-
SLOC(`Email`,`@`)),' ',1)))- +
(SLOC(SMOVE(`Email`,(SLOC(`Email`,`@`)+1),(SLEN(`Email`)-
SLOC(`Email`,`@`)),' ',1),'.')) >= 2
```

part of the rule.

Have fun implementing this cool code in your R:BASE applications!

Part

VI

6 Data Browser

6.1 Adding a Duplicate Row in the Data Browser/Editor

Did you know that you can add a duplicate row within the Data Browser/Data Editor?

Here's how:

While in the Data Browser/Editor, and while in "Edit Mode" [F4], an additional "Duplicate Row" option has been added to the Menu Bar.

Select "Edit" > "Duplicate Row" from the main Menu Bar to add a duplicate row.

In addition to the Menu Bar option, a "Duplicate Row" button has been added to the Toolbar and the key combination [Ctrl+F2] is also supported.

6.2 Conditionally Managing Data Browser/Editor Options

Traditionally, among many options, the Data Browser/Editor "Edit" list includes:

- Edit Mode [F4] (Switch Between Browse/Edit Mode)
- Print Data
- Export Results to
- Save Result As (Table, Temporary Table, View, Temporary View)

However, there may be circumstances when you may not want to provide these options to any or all users.

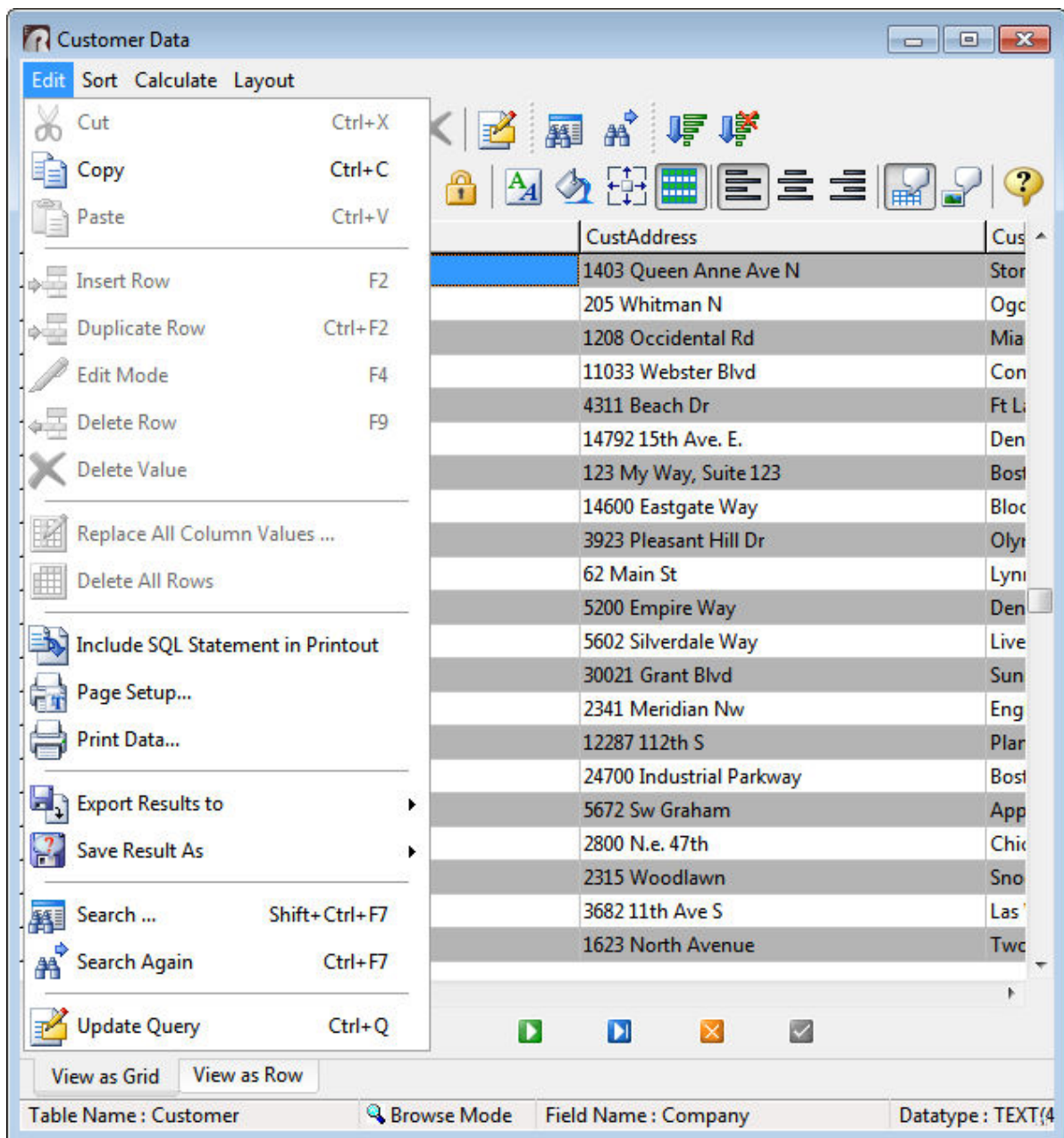
Did you know that you can conditionally manage these options which may include enabling/disabling these choices based upon the logged in user?

Here's how:

```
IF (CVAL('DATABASE')) <> 'RRBYW19' OR (CVAL('DATABASE')) IS NULL THEN
  CONNECT RRBYW19 IDENTIFIED BY NONE
ENDIF
```

```
-- Example 01 (Typical Options)
CLS
BROWSE Company,CustAddress,CustCity,CustState,CustZip,CustPhone +
FROM Customer ORDER BY Company +
OPTION CAPTION Customer Data|WINDOW_STATE NORMAL
RETURN
```

```
-- Example 02 (NOCHG)
-- Disable [F4] key to switch between Browse/Edit Mode
CLS
BROWSE Company,CustAddress,CustCity,CustState,CustZip,CustPhone +
FROM Customer ORDER BY Company NOCHG +
OPTION CAPTION Customer Data|WINDOW_STATE NORMAL
RETURN
```

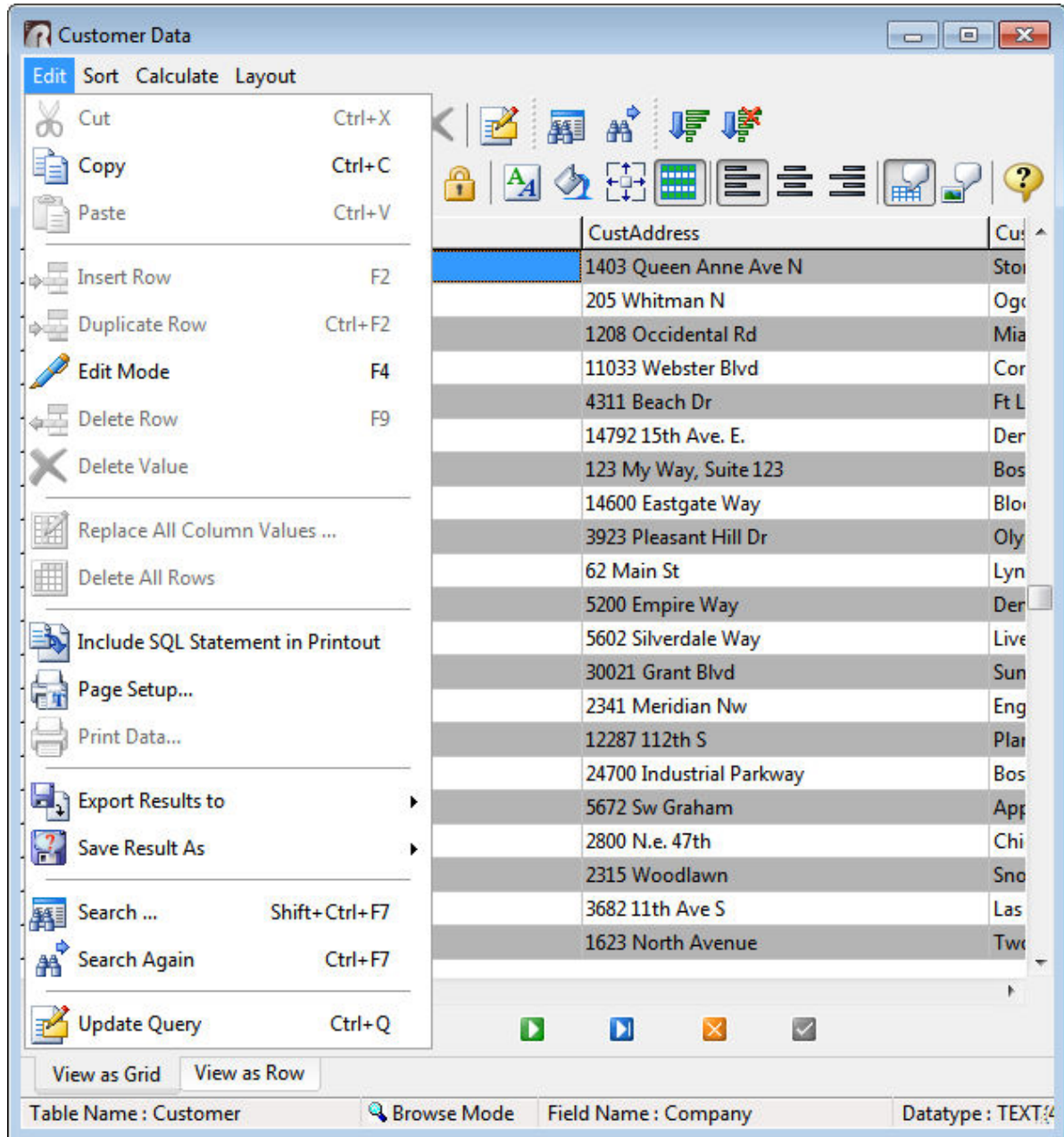
```
-- Example 03 (READONLY ON)
-- Makes the entire data set READONLY.
CLS
BROWSE Company,CustAddress,CustCity,CustState,CustZip,CustPhone +
FROM Customer ORDER BY Company +
OPTION CAPTION Customer Data|WINDOW_STATE NORMAL|READONLY ON
RETURN

-- Example 04 (PRINTDATA OFF)
-- Disable Print Data Option
CLS
```

```

BROWSE Company,CustAddress,CustCity,CustState,CustZip,CustPhone +
FROM Customer ORDER BY Company +
OPTION CAPTION Customer Data|WINDOW_STATE NORMAL|PRINTDATA OFF
RETURN

```

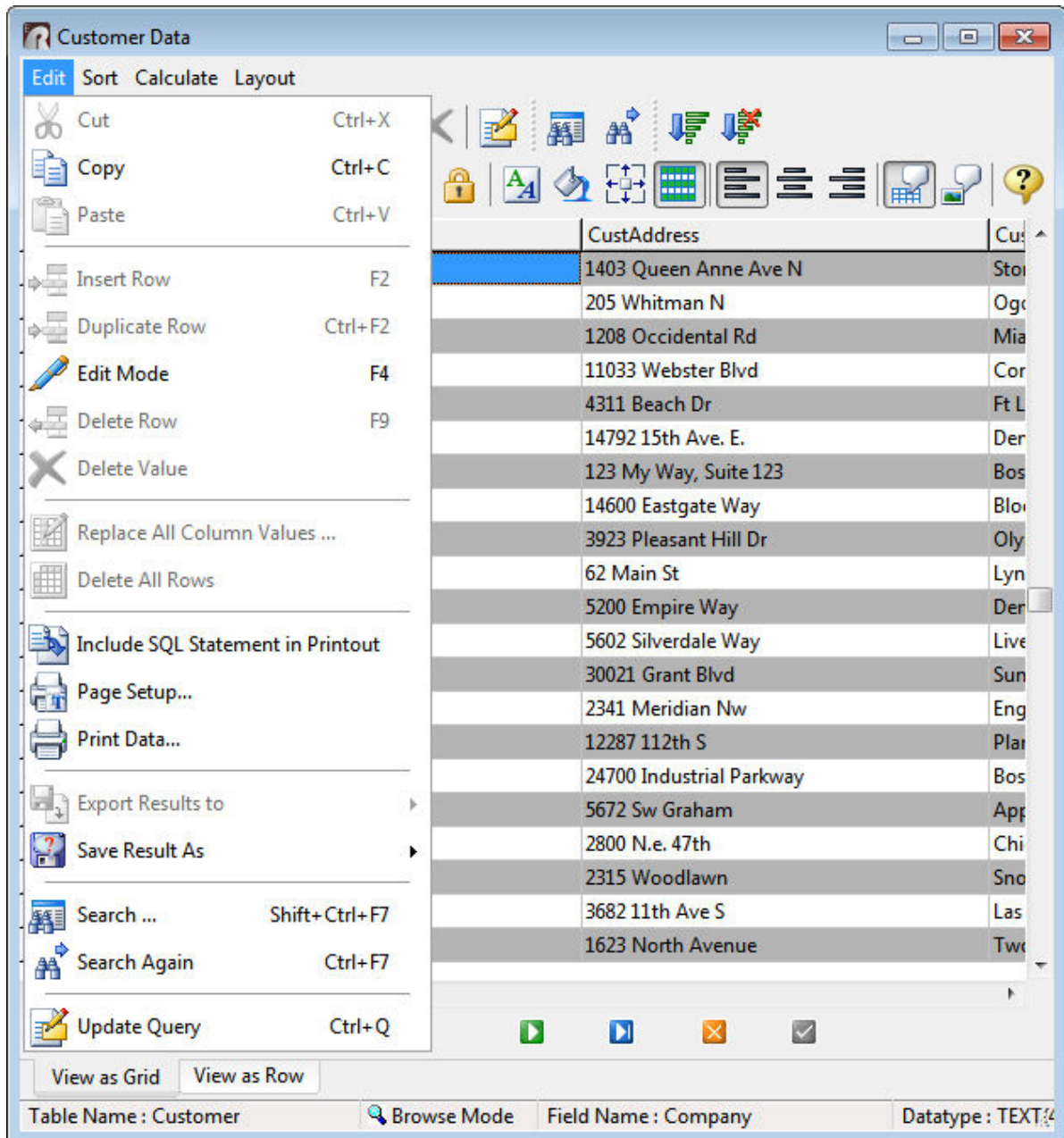


```

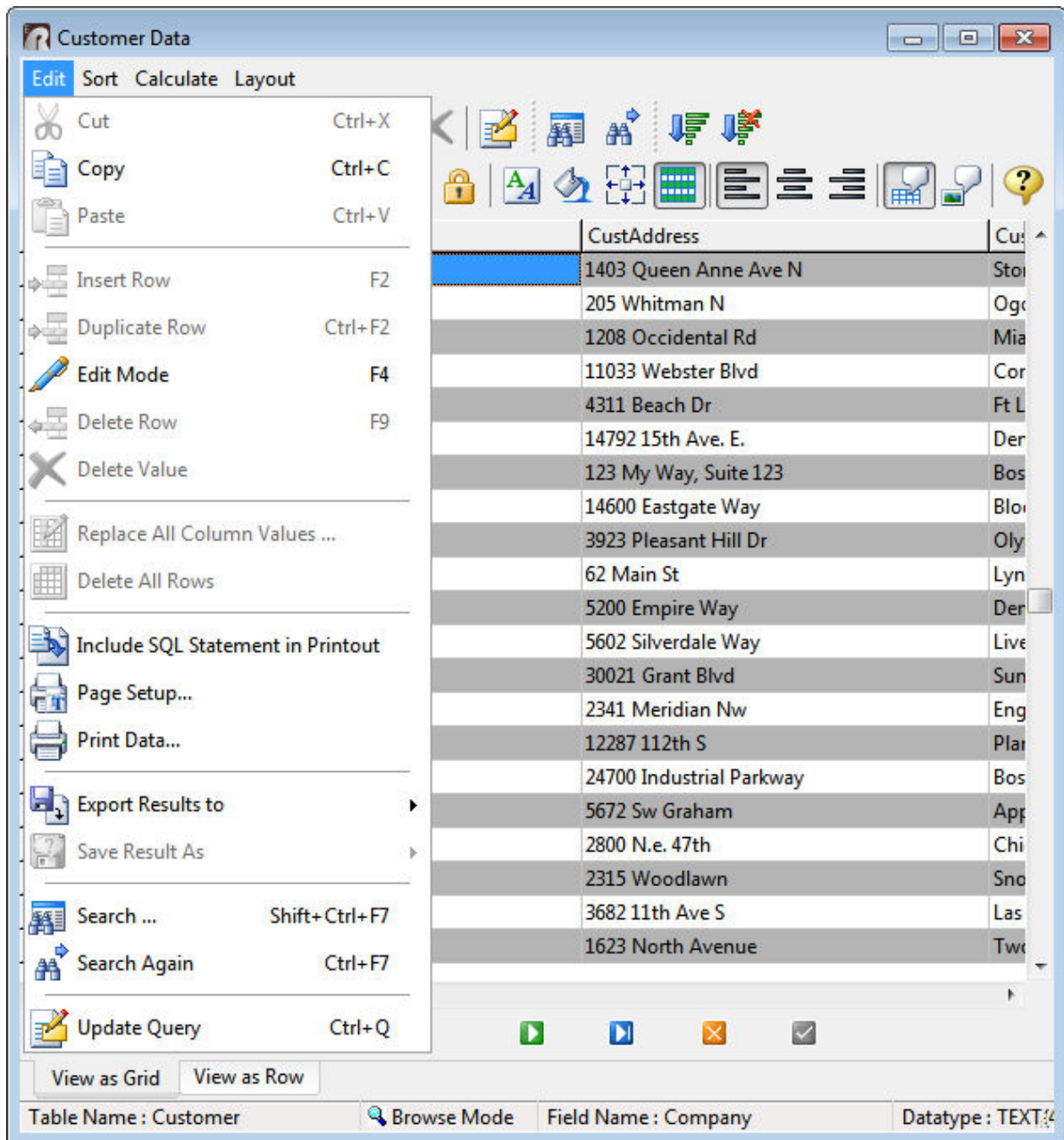
-- Example 05 (EXPORTRESULT OFF)
-- Disable Export Result To Option
CLS
BROWSE Company,CustAddress,CustCity,CustState,CustZip,CustPhone +
FROM Customer ORDER BY Company +
OPTION CAPTION Customer Data|WINDOW_STATE NORMAL|EXPORTRESULT OFF

```

RETURN



```
-- Example 06 (SAVERESULT OFF)
-- Disable Save Result As Option
CLS
BROWSE Company,CustAddress,CustCity,CustState,CustZip,CustPhone +
FROM Customer ORDER BY Company +
OPTION CAPTION Customer Data|WINDOW_STATE NORMAL|SAVERESULT OFF
RETURN
```

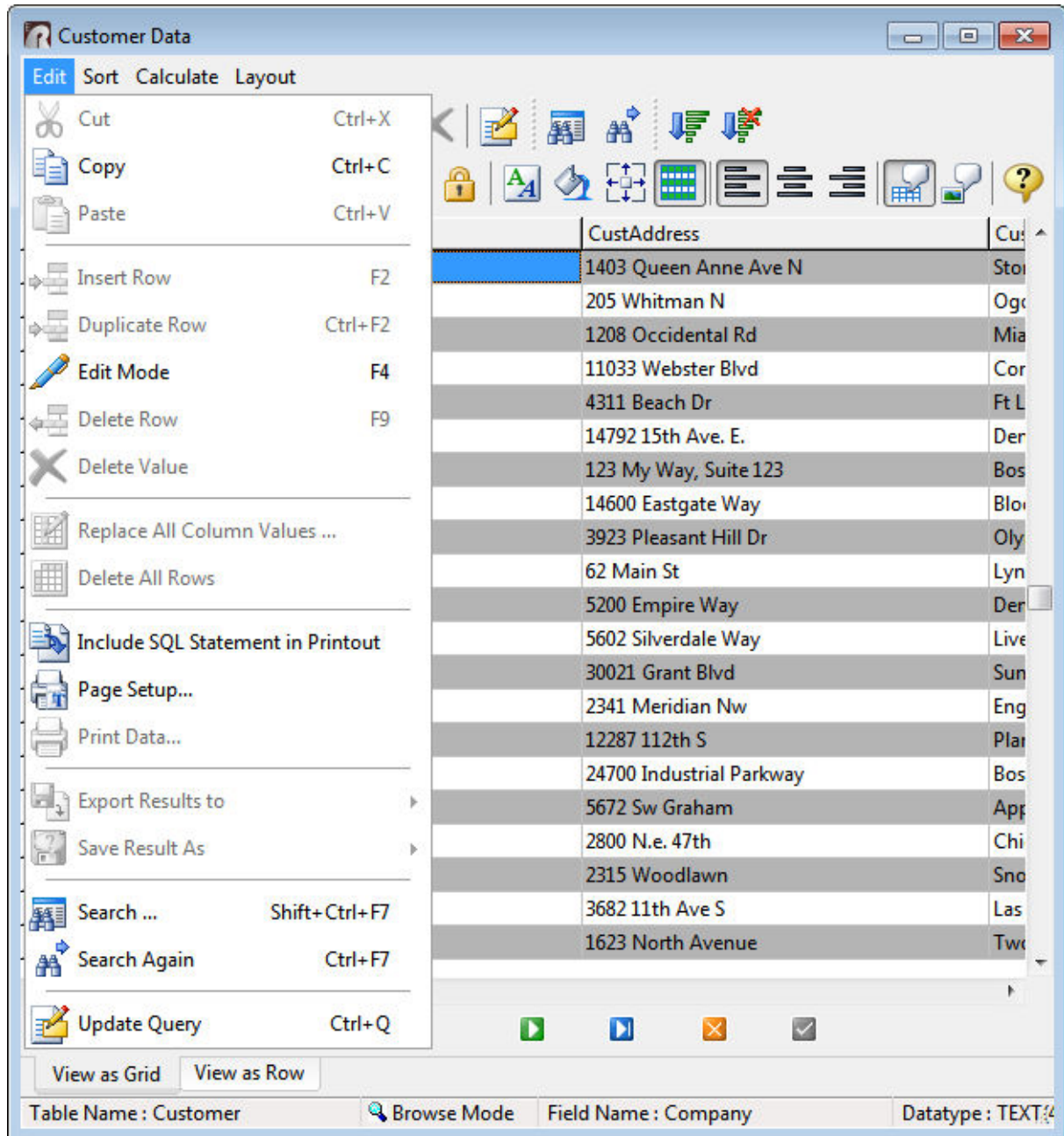


```
-- Example 07 (Conditionally Disable Options)
-- (NOCHG,READONLY ON,PRINTDATA OFF,EXPORTRESULT OFF,SAVERESULT OFF)
CLS
SET VAR vOptions TEXT = NULL
IF (CVAL('NetUser')) IN (JohnDoe,JaneDoe,MikeSmith,LLooser) THEN
    SET VAR vOptions = +
        ('CAPTION Customer Data|WINDOW_STATE NORMAL|READONLY ON' + +
         '|PRINTDATA OFF|EXPORTRESULT OFF|SAVERESULT OFF')
ELSE
    SET VAR vOptions = ('CAPTION Customer Data|WINDOW_STATE NORMAL')
ENDIF
BROWSE Company,CustAddress,CustCity,CustState,CustZip,CustPhone +
```

```

FROM Customer ORDER BY Company NOCHG +
OPTION &vOptions
CLEAR VARIABLES vOptions
RETURN

```



Have fun implementing these options in your R:BASE applications!

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

6.3 Data Browser Export Options

Did you know that you can configure the Data Browser to open a file after data is exported and to display blanks if the value is zero, just like the settings within the File Gateway utility?

Here's how:

Database Explorer > Main Menu > Settings > Data Browser > Additional Options > Export Options:

- Open After Export - launches the file created, based upon the current file associated program
- Blank If Zero - specifies to display all zero values as blanks fields

6.4 Right-Click Calculate (Compute, Tally, Duplicates)

Did you know that while in the Data Browser/Editor, using the right-click on selected column will bring up a new speed menu option to calculate the following three options?

Calculate >

- Compute...
- Tally...
- Duplicates..

Using the right-click on selected column, and selecting the Calculate option, now you can instantly Compute (Count, Rows, Minimum, Maximum, Sum, Average, Standard Deviation, Variance), Tally, and most importantly can find the Duplicates without knowing the correct syntax.

Part

VII

7 Stored Procedures

7.1 Order of Processing for Stored Procedures and Triggers

A trigger automatically runs a stored procedure when an UPDATE, DELETE, or INSERT command is run with a table.

The following describes the sequence in which data integrity is maintained when using Stored Procedures and Triggers:

DELETE

- Rules
- Cascades
- Triggers
- Keys
- Row itself

UPDATE

- Rules
- Cascades
- Triggers
- Keys
- Row itself

INSERT

- Rules
- Triggers
- Keys
- Row itself

Triggers automatically run a stored procedure when an UPDATE, DELETE, or INSERT command is run with a table. The trigger is run before the row that triggered it is updated, inserted, or deleted.

Since the trigger runs first, it gives you a chance to abort the procedure.

When indexes are updated it is the last step before completing the update.

The row has valid values, the rules are met, and the triggers did not abort.

The only error out at this point is if the key update fails, such as a PKFK violation. When the key processing code is done it is a real attempt to update the key, not just a "will it be ok" process. If you do not want your triggers to run you could modify the trigger to check for PK-FK violations before doing the rest of the trigger.

Part

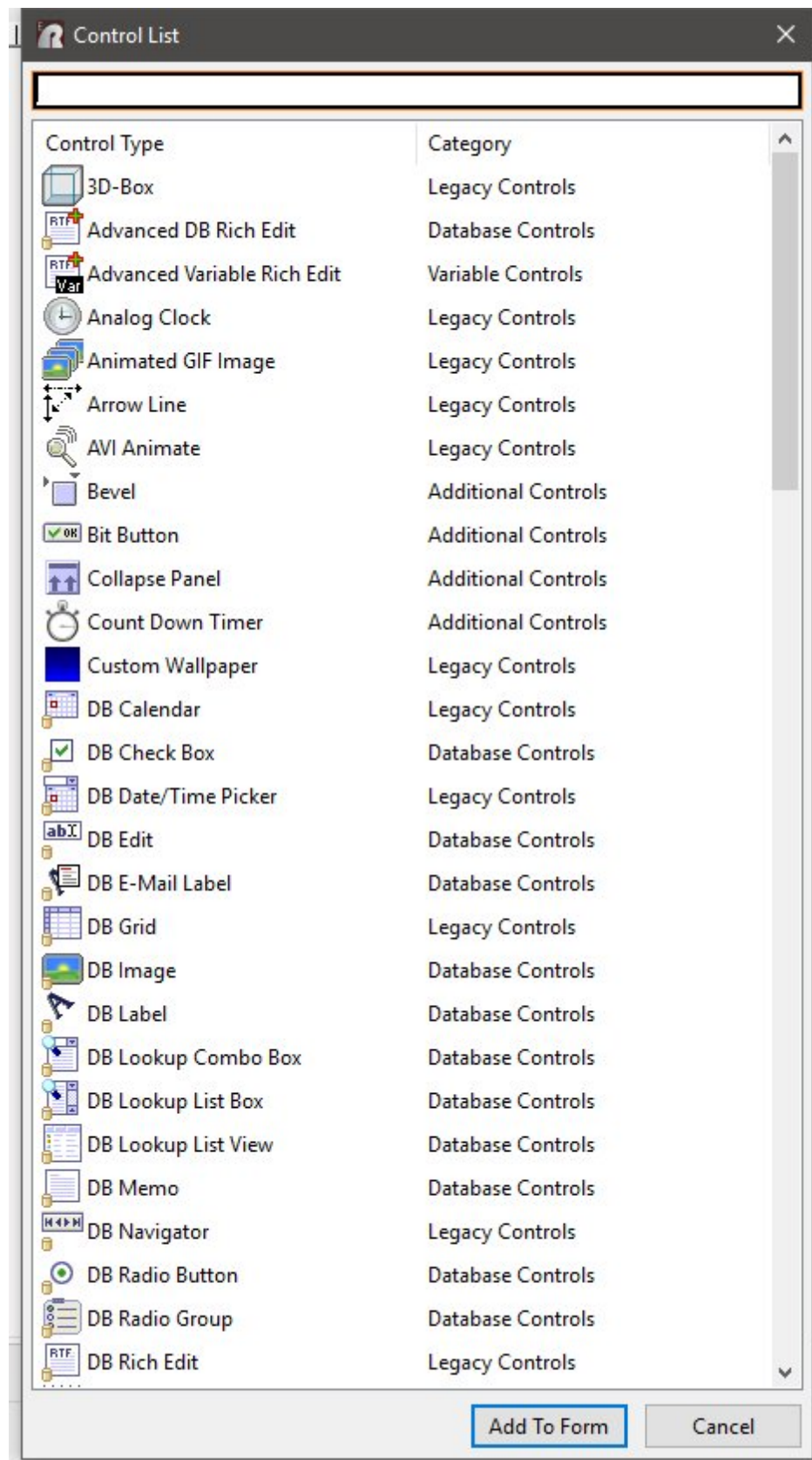
VIII

8 Forms

8.1 Accessing the Control List

Did you know that you can access the control list by pressing [Ctrl+Alt+P] keys in the Form Designer?

Then, you can see a complete list of all available controls, including the option to filter the list, as well as selecting the object and "Add To Form" button.



Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

8.2 Action Sequence (Form EEP Execution)

Action Sequence (the order in which the form EEPs are executed)

A typical control's property options within the "EEPs" tab include the following options:

- Predefined Actions
 - Add Row
 - Add Row and Exit
 - Delete Row
 - Discard Row
 - Discard Row and Exit
 - Duplicate Row
 - Exit
 - Next Row
 - Next Table
 - Previous Row
 - Previous Table
 - Save Row
 - Last Row
 - First Row
 - Save Row and Exit
 - Refresh Current Table
- Stored Procedure
- Custom Form Action
- Custom

This is the actual order in which the form EEPs are executed:

1. Custom Embedded EEP (if defined)
2. Custom EEP (External File)
3. Stored Procedure
4. Custom Form Action
5. Predefined Actions

8.3 Added Advanced Variable Rich Edit Control

Did you know that the new Advanced Variable Rich Edit control has been introduced in the latest versions of R:BASE?

Form Designer > Variable Controls > Advanced Variable Rich Edit

Using this control in forms, now you can take advantage of displaying and editing "formatted" NOTE or VARCHAR variable data.

If you have an existing Variable Rich Edit control placed within forms, you can easily convert control to an "Advanced Variable Rich Edit" as follows:

Within the Form Designer, right-click on the "Variable Rich Edit" and select the "Convert to Advanced Variable Rich Edit" option. Then, you can take advantage of all Properties, Effects, Scroll Bars, and EEPs for the Advanced Variable Rich Edit Control in Forms.

8.4 Additional Fields for DB/Variable Email Label Controls

Did you know that you can now use additional fields to specify the CC, BCC, and Message Body fields for the DB Email Label and Variable Email Label controls?

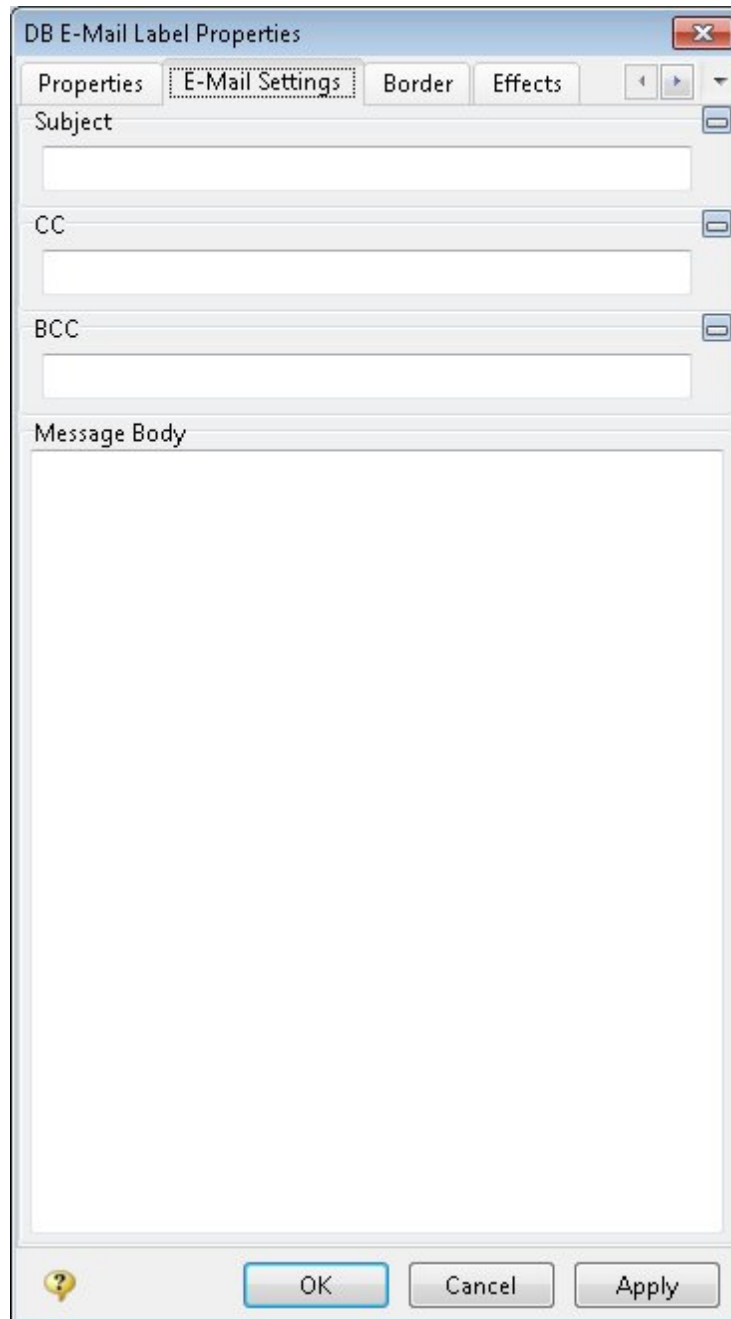
The Email Label Controls send a message to the user's MAPI mail client. The CC, BCC, and Message Body fields can be specified within the control properties, or altered on the fly using the PROPERTY command.

Example:

```
PROPERTY Comp_ID_CC CC 'joe@newco.com'
```

```
PROPERTY Comp_ID_BCC BCC 'wade@newco.com'
```

```
PROPERTY Comp_ID_MsgBody BODYTEXT 'Thank you for your order!'
```



The CC, BCC, and Message Body fields for the DB and Variable Email Label controls can now be specified using the PROPERTY command, and within the control properties!

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

8.5 Aligning Radio Groups Buttons

Typically, while in Form Designer, when you place a DB Radio Group or a Variable Radio Group control, by default the "Radio Buttons" are always aligned to the left side of the caption (Item Text).

Did you know that now you can change the Radio Button Alignment in conjunction with the associated Radio Button Caption (Item Text)?

A new "Button Alignment" property for DB and Variable Radio Group controls has been implemented to move the Radio Button on either side of the item text to provide you with a customized look & feel.

Form Designer > DB Radio Group > Object Property > Radio Buttons > Button Alignment ...

- Left Justify
- Right Justify

Form Designer > Var Radio Group > Object Property > Radio Buttons > Button Alignment ...

- Left Justify
- Right Justify

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

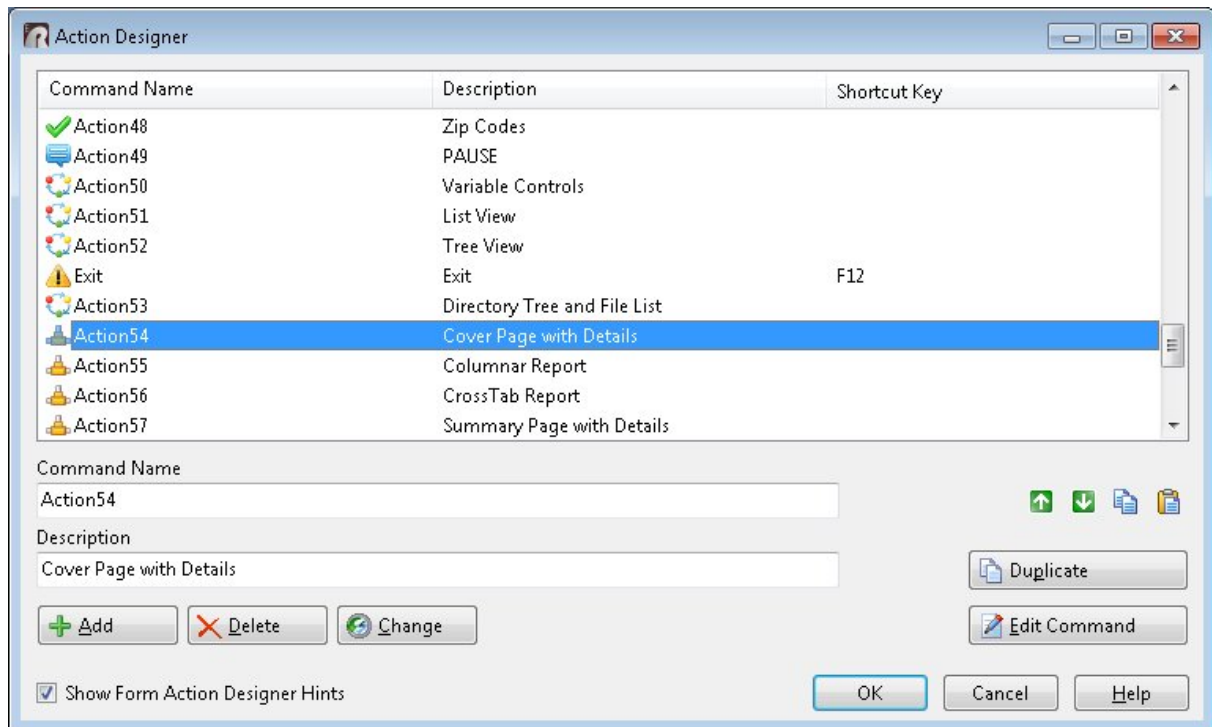
8.6 Assigning Keys to Execute Custom Form Actions

Did you know that you can assign a key, key combination, or a function key to execute any code defined as Custom Form Action in a form?

Custom Form Actions have been enhanced with keyboard shortcut support, to launch the action when a key or a key combination is pressed. Here's how ...

1. Start R:BASE
2. Connect the database and open a form in the Form Designer
3. Now define a Custom Form Action (procedure) that will be assigned a Shortcut Key.
4. Once the Custom Form Action is defined, right-click and select the Shortcut Key
5. Press any key or key combination to assign Shortcut Key, for example [Shift+Ctrl+C]
6. Click on [OK] button to save

Notice the Shortcut Key assigned to Custom Form Action

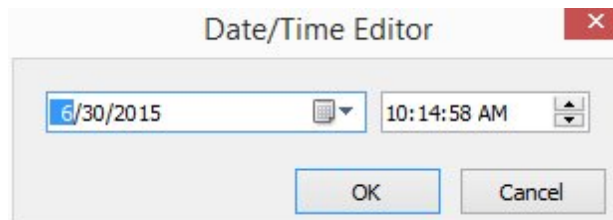


Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

8.7 Automating Date/Time Fields

Did you know... If you have a DB Edit or Variable Edit control placed on a form that is associated with a column or variable having DATETIME data type, you can automate the data entry by double-clicking or using the [Alt+Insert] to bring up the Date/Time Editor to fill-in the date time value.



The DB Edit/Variable Edit controls were enhanced to display a date/time editor, if the control is based upon a DATETIME data type. The date/time editor will open if the control is double clicked or if [Alt+Insert] is pressed.

If exists, the control hint will be used as the date/time editor title bar text.

Note:

The double click will only work if there is no EEP assigned to the double click event.

8.8 Capturing Column Name and Sort Order

Did you know that you can capture the column name and sort order of data within an Enhanced DB Grid, regardless of focus?

Additional GETPROPERTY and PROPERTY commands are implemented for Enhanced DB Grid Control.

Here's how ...

Syntax:

```
-- To programmatically capture the last column header clicked
  GETPROPERTY <EnhDBGridComponentID> CLICKHEADER 'varname'

-- To programmatically capture the sort order of sorted column
  GETPROPERTY <EnhDBGridComponentID> SORTORDER 'varname'
```

Possible return values are ASC and DESC.

If there is no sorted column, the value will be returned as DESC.

```
-- To programmatically emulate user click on column
  PROPERTY <EnhDBGridComponentID> CLICKHEADER 'EnhDBGridColumnName'
```

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

8.9 Capturing the Value of Form Component IDs

Did you know that you can capture the value of any DB or Variable Edit Control's Component ID, if defined, while using the form in Enter, Edit or Browse mode?

In addition to all Form System Variables, a new RBTI_FORM_COMPID variable has been added to capture the value of DB or Variable Edit Component ID, if defined.

Here's a comprehensive list of System Variables used in Forms:

- RBTI_DBGRID_COLUMN - Stores the name of focused Column on a DB Grid
- RBTI_DIRTY_FLAG - Returns 1 if any database control value(s) in the form was changed, or 0 if nothing was changed
- RBTI_FORM_ALIAS - Stores the name of a focused form, if AS alias is used
- RBTI_FORM_CLICKED - Stores to the clicked control's Component ID property
- RBTI_FORM_COLNAME - Stores the name of focused column of a database control on a form
- RBTI_FORM_COLVALUE - Stores the value of focused column database control on a form
- RBTI_FORM_COMPID - Stores the value of focused DB/Variable Edit control's Component ID
- RBTI_FORM_CTRLTEXT - Stores to the clicked control's "Caption" or "Text" property
- RBTI_FORM_DATATYPE - Stores the data type of focused column database control on form
- RBTI_FORM_DBLCLKED - Stores the double clicked control's Component ID property
- RBTI_FORM_DSGN - Stores the value of "YES" when a form is open in the Form Designer
- RBTI_FORM_DIRTYVAR - Returns 1 if any Variable Control value(s) in form were changed, or 0 if nothing was changed
- RBTI_FORM_EXT_DSGN - holds the value of "YES" when a form is open in the External Form Designer
- RBTI_FORM_FORMNAME - Stores the name of the current form (useful when using multiple forms simultaneously)
- RBTI_FORM_MODE - Stores the value of current mode of the form; ENTER, EDIT, or BROWSE
- RBTI_FORM_TBLNAME - Stores the name of the current table in a form (useful with multi-table forms)
- RBTI_FORM_VARNAME - Stores the name of focused Variable Control on a form
- RBTI_FORM_VARVALUE - Stores the value of focused Variable Control on a form

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

8.10 Choosing the Right Form Control

With the growing community of R:BASE users migrating their legacy DOS databases and applications to R:BASE for Windows, it can be overwhelming when upgrading and/or designing forms.

There are so many controls and so many options.

At times, application developers may be unsure what control/object to add to a form, when end users are requesting a menu to select from a list of preset values.

Based upon the how the list of information is to be displayed, where the list of available data resides, and if the selected item will be stored or only captured, there are different form control types that can be used to display the perfect option to the users.

Control Display

The type of control to add to a form depends on the amount of available space on the form and how the list of information is to be presented to the users.

When limited space is available on forms, a "Combo Box" control is ideal, as the list of options to select is "dropped down" and visible when the control is selected.

If space is not a factor, "List Box" and "List View" controls are available to present a box-type object with a list of selections initially displayed. The List Box presents the list of data to select in a single column, and perhaps separated. The List View presents a list of data selections in multiple columns within a grid-like display. The objects consume a larger amount of space but can be sized accordingly.

When the content within the Combo Box, List Box, or List View is larger than visible listing, a scroll bar is available to browse further content.

Source of Data to Select

The source of data populating the control will determine the type of control. If the source is stored within a table, a "Lookup" control is to be used. Otherwise, if the list of options will be manually added and stored within the control, a "User Defined" control is used. A User Defined control can contain a small selection of items (e.g. Yes, No, Maybe) or a large list. In most cases, a "Lookup" control is assigned.

End Result for Selection

Another factor to decide what control to use is the "end result" to take after the item, or items, is selected by the user. If the selection is to be stored directly into a table, the "DB" control is used. If the selection must be captured into a variable, use the "Variable" type control.

Controls and Examples of Use

DB Lookup Combo Box - Displays a drop-down menu list of data pulled from a table. The selection is stored in another table. An example is a list of job titles that can be assigned to employees. The titles are stored in a different table than the employees.

DB Lookup List Box - Displays a box of listed data pulled from a table. The selection is stored in another table. An example is a list of vendors that can be assigned to an order. The vendors are stored in a different table than the orders.

DB User Defined Combo Box - Displays a drop-down menu list of options defined and stored within the control. The selection is stored in a table. An example is a drop-down menu of predefined answers for a questionnaire.

DB User Defined List Box - Displays a box of listed data defined and stored within the control. The selection is stored in a table. An example is a full displayed list of predefined answers for a questionnaire.

DB Lookup List View - Displays a grid of listed data pulled from a table. The selection is stored in a table. An example is a grid of companies, with the city and state aligned in separate columns, that can be assigned to an order. The company data stored in a different table than the orders.

Variable Lookup Combo Box - Displays a drop-down menu list of data pulled from a table. The selection is captured into a defined variable. An example is a drop-down menu of employees where the selection is used in further form processing.

Variable Lookup List Box - Displays a box of listed data pulled from a table. The selection is captured into a defined variable. An example is a list of dates where transactions occurred for a specific month where the selection is used in further form processing.

Variable User Defined Combo Box - Displays a drop-down menu list of options defined and stored within the control. The selection is captured into a defined variable. An example is a drop-down menu of print output options used in report processing.

Variable User Defined List Box - Displays a box of listed data defined and stored within the control. The selection is captured into a defined variable. An example is a list of available reports used in report processing.

Variable Lookup List View - Displays a grid of listed data pulled from a table. The selection is captured into a defined variable. An example is a grid of transactions, with details of quantity, prices, etc., aligned in separate columns. The selection is used in further form processing.

Additional Controls

Many other control types are available and can be added to R:BASE forms based upon the data presented to users. Be sure to review the other available controls to provide the best available appearance of the displayed data.

8.11 Clearing the PDF Viewer Document

Did you know that you can clear the document that is being displayed in the PDF Viewer Control?

A new PROPERTY command has been introduced support the functionality.

Syntax:

```
PROPERTY PDFViewerCompID CLEAR 'TRUE'
RETURN
```

The property "CLEAR" and the value "TRUE" will clear the previously loaded file, if any.

Contrary to PROPERTY command, the GETPROPERTY PDFViewerCompID CLEAR varname command will return the name of a previously cleared document, if any.

Syntax:

```
GETPROPERTY PDFViewerCompID CLEAR varname
RETURN
```

Here is an example to demonstrate the use of this cool property on forms with the PDF Viewer control.

```
-- Load PDF Document Button
-- Author: A. Razzak Memon
PROPERTY PDFVIEWER FILENAME 'CommandIndex.pdf'
SET VAR vPDFPageNumber TEXT = NULL
SET VAR vPDFPageCount TEXT = NULL
SET VAR vPDFPage TEXT = NULL
GETPROPERTY PDFViewer PAGENUMBER vPDFPageNumber
GETPROPERTY PDFViewer PAGECOUNT vPDFPageCount
SET VAR vPDFPage = (.vPDFPageNumber &'of'&.vPDFPageCount)
PROPERTY LblPages CAPTION .vPDFPage
PROPERTY PDFViewer ZOOM '65'
PROPERTY RBASE_FORM_ACTION CheckZoom ''
CLEAR VAR vPDFPageNumber, vPDFPageCount, vPDFPage
RETURN
```

```
-- Clear PDF Viewer Button
-- Author: A. Razzak Memon
```

```
PROPERTY PDFViewer CLEAR 'TRUE'
RETURN
```

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

8.12 Click-Based RBTI System Variables

Three click-based RBTI System Variables were introduced to capture information with mouse events:

- RBTI_FORM_DBLCLKED
- RBTI_FORM_CLICKED
- RBTI_FORM_CTRLTEXT

During the form runtime mode (ENTER USING, EDIT USING, BROWSE USING), if a control is "double-clicked", the RBTI_FORM_DBLCLKED system variable will be set to the clicked control's "Component ID" property (if exists). The value can then be used by the PROPERTY/GETPROPERTY commands to manipulate the control.

During the form runtime mode (ENTER USING, EDIT USING, BROWSE USING), if a control is "clicked", the RBTI_FORM_CLKED system variable will be set to the clicked control's "Component ID" property (if exists). The value can then be used by the PROPERTY/GETPROPERTY commands to manipulate the control.

During the form runtime mode (ENTER USING, EDIT USING, BROWSE USING), if a control is "clicked", the RBTI_FORM_CTRLTEXT system variable will be set to the clicked control's "Caption" or "Text" property.

Imagine the possibilities and use of these new RBTI System Variables in your R:BASE applications!

8.13 Controlling Date/Time Picker Drop-Down

Did you know that you can control the drop-down option of the DB Date/Time Picker or Variable Date/Time Picker control by simply using the following PROPERTY command?

```
PROPERTY <DateTimePickerCompID> OPEN 'TRUE'
RETURN
```

That will open the DB/Variable Date/Time Picker drop-down calendar, on demand.

8.14 Controlling TABLE Add/Delete Row Properties

Did you know that you can control the "Add" or "Delete" Rows options for the table settings within a Form using the following PROPERTY command parameters?

- ENABLE_ADD_NEW_ROWS
- DISABLE_ADD_NEW_ROWS
- ENABLE_DELETE_ROWS
- DISABLE_DELETE_ROWS

Examples:

```
-- To enable the option for "Add New Rows to the Table":
PROPERTY TABLE tablename ENABLE_ADD_NEW_ROWS
RETURN
```

```
-- To disable the option for "Add New Rows to the Table":
PROPERTY TABLE tablename DISABLE_ADD_NEW_ROWS
```

```
RETURN
```

```
-- To enable the option for "Delete Rows":
```

```
PROPERTY TABLE tablename ENABLE_DELETE_ROWS
```

```
RETURN
```

```
-- To disable the option for "Delete Rows":
```

```
PROPERTY TABLE tablename DISABLE_DELETE_ROWS
```

```
RETURN
```

8.15 Converting DB Grid Control to Enhanced DB Grid Control

Did you know that you can convert an existing DB Grid Control to Enhanced DB Grid Control by click of a mouse button?

Here's how:

While in Form Designer, right-click on an existing DB Grid Control and then select "Convert to Enhanced DB Grid" from the list of speed menu options.

The following single-click conversions to "enhanced" form controls are also available in R:BASE:

- Panel -> Convert to Enhanced Panel
- Group Box -> Convert to Enhanced Group Box
- Speed Button -> Convert to Enhanced Speed Button
- Tab Control -> Convert to Enhanced Tab Control
- Calendar -> Convert to Enhanced Calendar
- Variable Calendar -> Convert to Enhanced Variable Calendar
- DB Calendar -> Convert to Enhanced DB Calendar
- DB Grid -> Convert to Enhanced DB Grid
- DB Navigator > Convert to Enhanced DB Navigator
- DB Rich Edit -> Convert to Advanced DB Rich Edit
- Speed Button -> Convert to Enhanced Speed Button

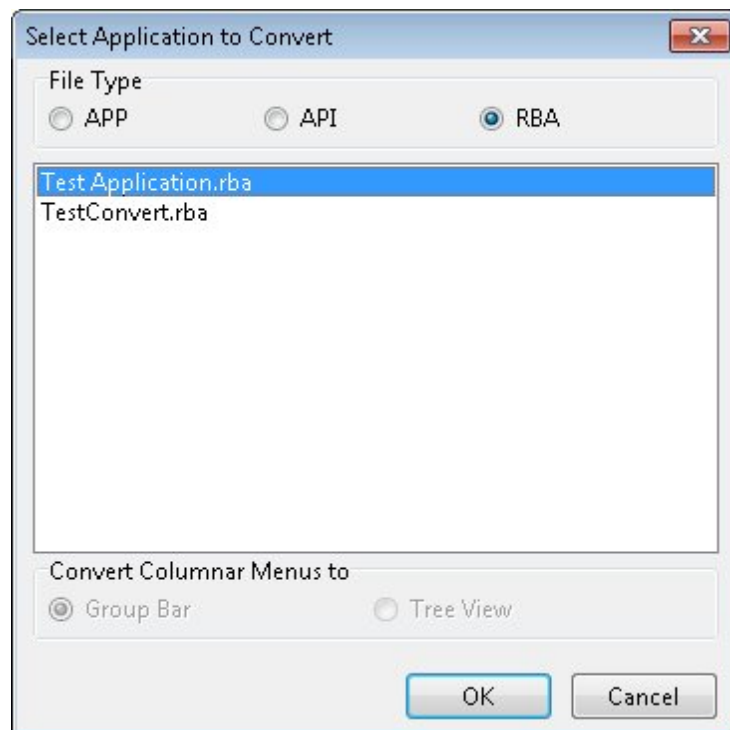
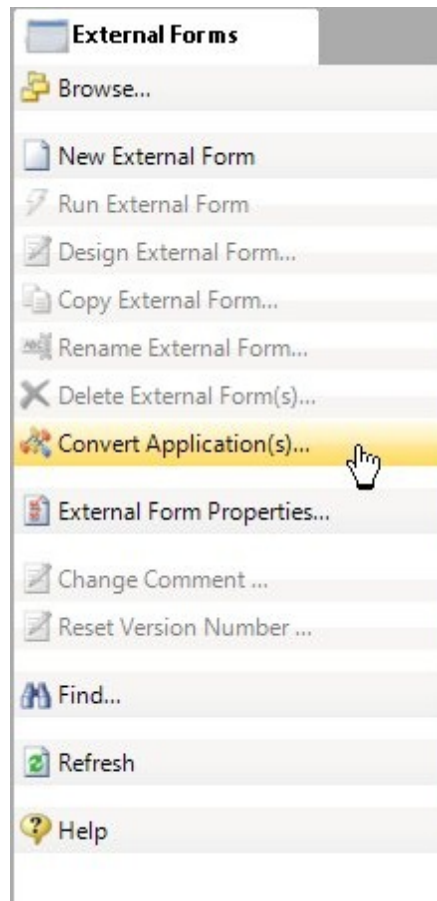
Supported Versions:

- R:BASE X (Build..: 10.0.1.10522 or higher)

8.16 Converting R:BASE Application Files to External Form Files

Did you know that you can convert an R:BASE Application Files (.RBA) to an External Form Files (.RFF)?

Within the External Form File portion of the Database Explorer, the new "Convert Application(s)..." option is available to convert R:BASE Application Files to an External Form File.



Selecting this option will create an external form that matches the same properties used within the RBA application file.

When the external form file is created, the "Main Menu" section is converted to a form "Menu Bar", the "Actions" are converted to "Custom Form Actions", and the "Toolbar" is converted to a Toolbar in the new "Toolbar Designer".

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

8.17 Converting Variable Form to External Form File

Did you know that you can convert a Variable Form to External Form File?

A feature in the R:BASE Database Explorer within the "Forms" options allows for a Variable Form to be saved as an External Form File.

This option is only enabled when a Variable Form is selected.

Once the Variable form is highlighted, using the right-click speed menu, or clicking on the "Save as External Form File" option. A dialog box will prompt for the External Form File name with the .rff file extension.

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

8.18 Copy and Paste Custom Form Actions

Did you know that you can copy a Custom Form Action from a form, and paste it into another form?

Buttons for "Copy" and "Paste" have been added to the Action Designer for the Forms and External Form File modules, to allow users to copy and paste selected, or all, Custom Form Actions.

Please note the "Copy" and "Paste" buttons are located next to the "Up" and "Down" arrow buttons, which reorder the location of a Custom Form Action.

In addition, you may also use [Ctrl+C] to copy selected action(s) and [Ctrl+V] to paste.

Custom Form Action can now be easily moved from one form to another!

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

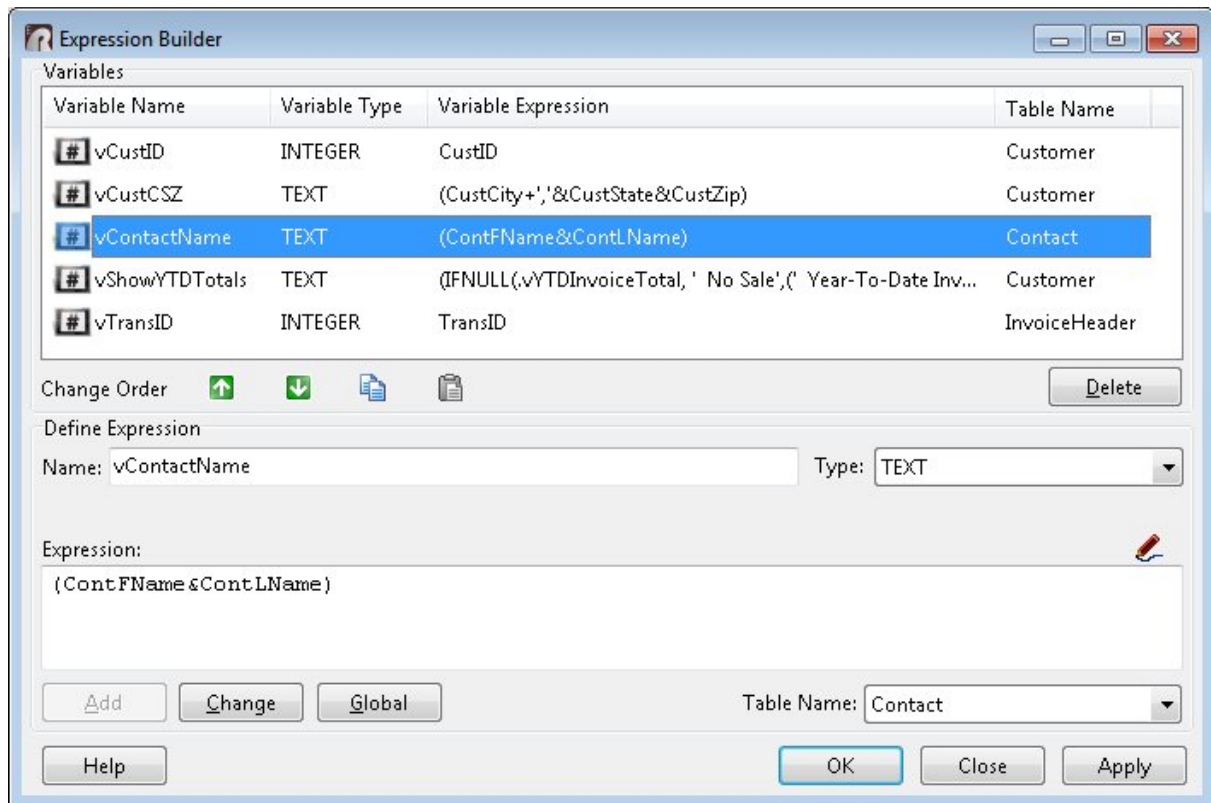
8.19 Copy and Paste Expression Builder Variables

Did you know that you can copy a form variable from the Expression Builder, and paste it into another form?

Buttons for "Copy" and "Paste" have been added to the Expression Builders for the Form, Report, and Label modules, to allow users to copy and paste selected, or all, variables.

Please note the "Copy" and "Paste" buttons are located next to the "Up" and "Down" arrow buttons, which reorder the location of a variable in the Expression Builder.

In addition, you may also use [Ctrl+C] to copy selected expression(s) and [Ctrl+V] to paste expression(s) while placing the cursor over the expression builder window.



Variables and Expressions can now be easily moved from form to form, label to label, and report to report!

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

8.20 Copy and Paste Menu Bar Items

Did you know that you can copy a Menu Bar Item from a form, and paste it into another form?

Buttons for "Copy" and "Paste" have been added to the Menu Bar Designer for the Forms and External Form File modules, to allow users to copy and paste selected, or all, items.

In addition, you may also use [Ctrl+C] to copy selected item(s) and [Ctrl+V] to paste.

Menu Bar Items can now be easily moved from one form to another!

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

8.21 Copy and Paste Tree View Nodes

Did you know that you can copy and paste nodes within the Tree View control?

Buttons for "Copy" and "Paste" have been added to the Tree View control to allow users to copy and paste nodes. When a node is copied, all sub-items are copied with the main node. Nodes can also be copied from one form to another.

In addition, you may also use [Ctrl+C] to copy the selected node and [Ctrl+V] to paste.

Tree View Nodes can now be easily moved into the same Tree View, or into another form!

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

8.22 Copying and Pasting Windows Clipboard Data in Forms

Did you know that you can copy and paste any image or text to and from the Windows Clipboard to any designated control on a form using the PROPERTY command?

The following PROPERTY command parameters are now supported:

- CopyToClipboard
- CutToClipboard
- PasteFromClipboard

Here's how:

- To copy images or text data to the Windows Clipboard

```
PROPERTY <ComponentID> CopyToClipboard 'TRUE'
RETURN
```

Supported Form Controls:

- DB Edit
- DB Memo
- DB Image
- Variable Edit
- Variable Memo
- Variable Image

2. To cut images or text data to the Windows Clipboard

```
PROPERTY <ComponentID> CutToClipboard 'TRUE'
RETURN
```

Supported Form Controls:

- DB Edit
- DB Memo
- DB Image
- Variable Edit
- Variable Memo
- Variable Image

3. To paste images or text data from the Windows Clipboard

```
-- Example 01
-- To paste Windows Clipboard BLOB Data into DB/VAR Image Field
PROPERTY <BLOBControlComponentID> PasteFromClipboard 'TRUE'
RETURN
```

```
-- Example 02
-- To paste Windows Clipboard Text Data into DB/VAR Edit/Memo Field
PROPERTY <EditMemoControlComponentID> PasteFromClipboard 'TRUE'
RETURN
```

```
-- Example 03
-- To paste Windows Clipboard BLOB Data into focused DB/VAR Image Field
```



```

PROPERTY RBASE_FORM PasteFromClipboard 'TRUE'
RETURN

-- Example 04
-- To paste Windows Clipboard Text Data into focused DB/VAR Edit/Memo
Field
PROPERTY RBASE_FORM PasteFromClipboard 'TRUE'
RETURN

```

8.23 Creating a Dynamic Tree View Object

Did you know that you can create a dynamic Tree View object in forms using the CREATEOBJECT command?

Syntax:

```
CREATEOBJECT <FormObjectName> 'ComponentID' 'Parent' T L W H
```

Here's an example to create a dynamic Tree View object with Node Items and Sub-Items.

```

-- Start here
-- Create TreeView Object
CREATEOBJECT TreeView 'TreeView' 'RBASE_FORM' 5 10 186 441
PROPERTY TreeView INIT 'TRUE'

-- Add Nodes (Items and Sub-Items)
-- Create First Item
CREATEOBJECT TREENODE Node1 TreeView 0 0 0 0
PROPERTY Node1 TEXT 'Node 1'
PROPERTY Node1 INIT 'TRUE'

-- Add Sub-Item to First Item
CREATEOBJECT TREENODE Node11 TreeView|Node1 0 0 0 0
PROPERTY Node11 TEXT 'Node 1 - 1'
PROPERTY Node11 INIT 'TRUE'

-- Create Second Item
CREATEOBJECT TREENODE Node2 TreeView 0 0 0 0
PROPERTY Node2 TEXT 'Node 2'
PROPERTY Node2 INIT 'TRUE'

-- Add Sub-Item to Second Item
CREATEOBJECT TREENODE Node21 TreeView|Node2 0 0 0 0
PROPERTY Node21 TEXT 'Node 2 - 1'
PROPERTY Node21 INIT 'TRUE'
RETURN
-- End here

--To delete the dynamically created Tree View object
DELETEOBJECT TreeView
RETURN

```

8.24 Customized Column Format in Variable Lookup List Views

Did you know that you can customize the column format in Variable Lookup List Views?

Here's an example using the "AllColumns" table in the RRBYW19 database:

01. Variable Lookup List View > Properties > Lookup Settings
02. Select the List Source ...
 - Table Name:
 - Key Field
 - Multi-Column
03. Columns Chosen:
 - CInteger,CReal,CCurrency,CDouble,(FORMAT(CDouble,'99999999.00'))
04. Columns ...
 - Column 1 (Integer Column)
 - Column 2 (Real Column)
 - Column 3 (Currency Column)
 - Column 4 (Double Column)
 - Column 5 (Double Column (Formatted))

Once all set and done, your form with Variable Lookup List View should look like this ...

Integer Column	Real Column	Currency Column	Double Column	Double Column (Formatted)
123456789	12345.79	\$10,999,999.00	12345.1289011	12345.13
567890123	12345.79	\$12,999,999.00	12345.1489011	12345.15
678901234	12345.79	\$13,999,999.00	12345.1589011	12345.16
789012345	12345.79	\$14,999,999.00	12345.1689011	12345.17
890123456	12345.79	\$15,999,999.00	12345.1789011	12345.18
901234567	12345.79	\$16,999,999.00	12345.1889011	12345.19
112345678	12345.79	\$17,999,999.00	12345.1989011	12345.20
123456789	12345.79	\$18,999,999.00	12345.2089011	12345.21
134567890	12345.79	\$19,999,999.00	12345.2289011	12345.23
145678901	12345.79	\$20,999,999.00	12345.2389011	12345.24
178901234	12345.79	\$22,999,999.00	12345.2689011	12345.27
456789012	12345.79	\$11,999,999.00	12345.1389011	12345.14
123456789	12345.79	\$18,999,999.00	12345.2189011	12345.22
156789012	12345.79	\$21,999,999.00	12345.2489011	12345.25
167890123	12345.79	\$21,999,999.00	12345.2589011	12345.26
189012345	12345.79	\$24,999,999.00	12345.2789011	12345.28

[Close](#)

Supported Versions:

- R:BASE X (Build..: 10.0.1.10727 or higher)

8.25 Customizing Character Case for Data Input

The R:BASE Form and External Form Designers include an option to customize the Character Case for DB and Variable Edit Controls.

Form Designer > DB Edit Control > Properties > Effects > Character Case

Form Designer > Variable Edit Control > Properties > Effects > Character Case

Did you know that you can customize the Character Case for DB and Variable Edit Controls using the following options?

- Normal (the default)
- Upper Case
- Lower Case
- Title Case
- Sentence Case

Character Case specifies a case of the entered characters, which are converted as the user types. Changing the Character Case property changes the actual contents of the text, not just the appearance.

Use "Upper Case" option to capitalize every character.

Use "Lower Case" option to not capitalize any character.

Use "Title Case" option to capitalize the first letter of every word.

Use "Sentence Case" option to capitalize the first letter of the sentence.

8.26 Customizing DB Grid Column Properties

By default, placing the DB Grid control on the form and assigning the appropriate table will use each column to be included in the DB Grid when displayed. However, if you wish to include only a selected column, you'll have to use the [Columns] option of the DB Grid Properties.

Once you select specific columns for DB Grid in selected columns panel, you can individually customize each column property by highlighting the column and then clicking on the "gear" icon. Using this option, you may customize the following Column Properties:

- General (Column/Title Properties)
- Pre-Defined Values
- Expression

You'll have to go through each column, one at a time, to customize individual column properties.

However, did you know that you can select more than one column (using Shift+Click, Click+Ctrl+Click, or Ctrl+Click), and then use the Right-Click to change the following column properties all at once?

Using that approach, you can customize the following column properties:

- Title Properties (Alignment, Color, Font)
- Column Properties (Alignment, Color, Font)
- Set Read Only ON
- Set Read Only OFF

For samples of customized DB Grid Columns, take a look at the following forms included within the RRBYW19 sample database.

- DBGridListBox
- DBGridLookupList
- DBGridProperties
- DBGridWithPickList
- SalesTransactions

Summary:

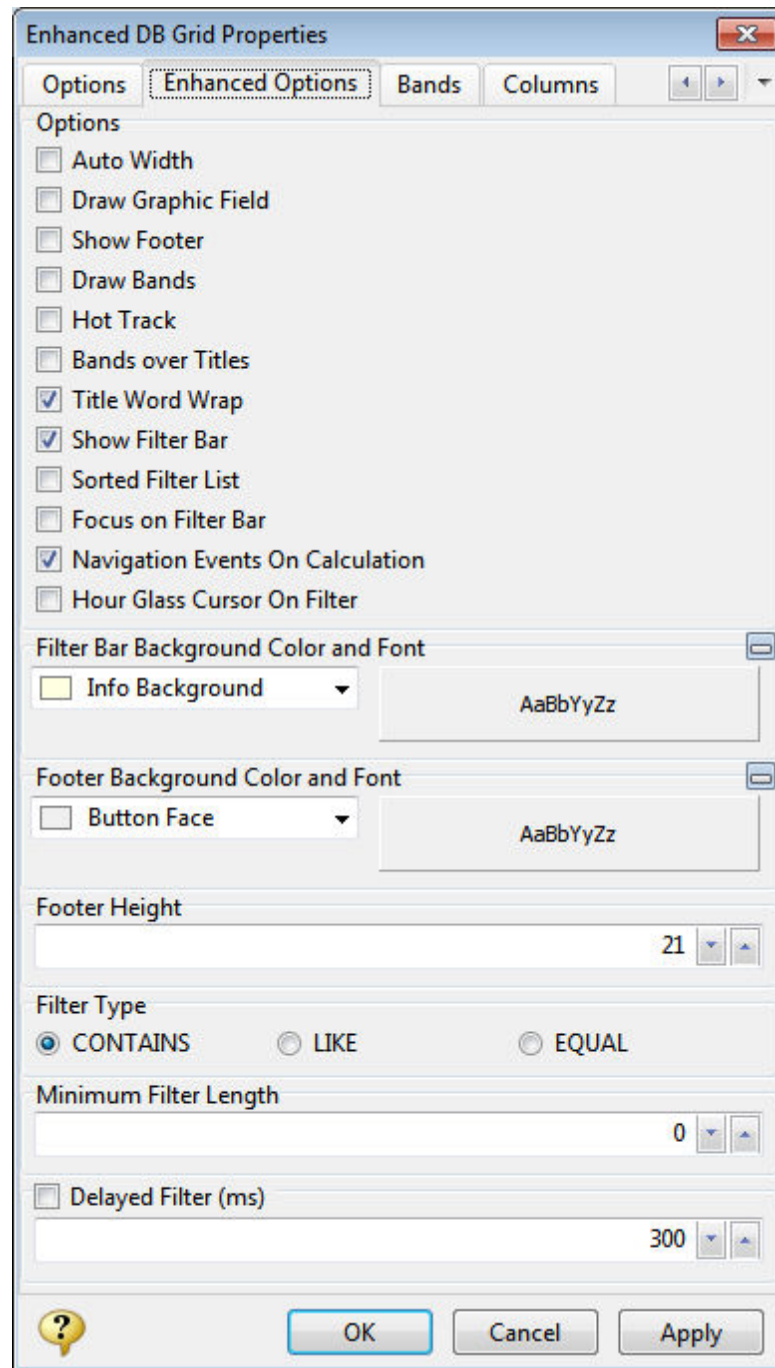
Highlighting all columns and then clicking on the "gear" icon will only modify the properties of "Last" column. However, if you wish to modify the properties of all selected columns, you'll need to use the right-click and then select the appropriate property from the speed menu.

8.27 Customizing Filter Options for the Enhanced DB Grid Control

When using the Enhanced DB Grid Control and selecting the option to "Show Filter Bar", the filter type default is always set to "CONTAINS".

Did you know that you can customize the filter option to use additional conditions such as "LIKE" or "EQUAL"? Here's how:

Form Designer > Enhanced DB Grid Control > Properties > Enhanced Options ...



8.28 Customizing the Form Closing Behavior

When closing table driven forms where data has been changed or deleted, a dialog will appear prompting "Save Changes?" or "Confirm Deletion?"

Did you know that, in addition to the "DONTSHOWSAVEDIALOG" and "DONTSHOWDELETEDIALOG" PROPERTY command parameters, you can use the "Don't Show Save Dialog" and "Don't Show Delete Dialog" attributes listed within the Form Settings?

Don't Show Save Dialog - disables the display of the "Save Changes?" dialog box upon closing the

form

Don't Show Delete Dialog - disables the display of the "Confirm Deletion?" dialog box upon closing the form

Using these settings, you can customize how table driven forms close.

8.29 Customizing the Form Menu Bar Appearance

The Menu Bar Designer supports the ability to add a custom menu bar across the top of forms. The Menu Bar uses main "Items" which display across the top of a window, and "Sub-Items" which drop down when the main Item is selected.

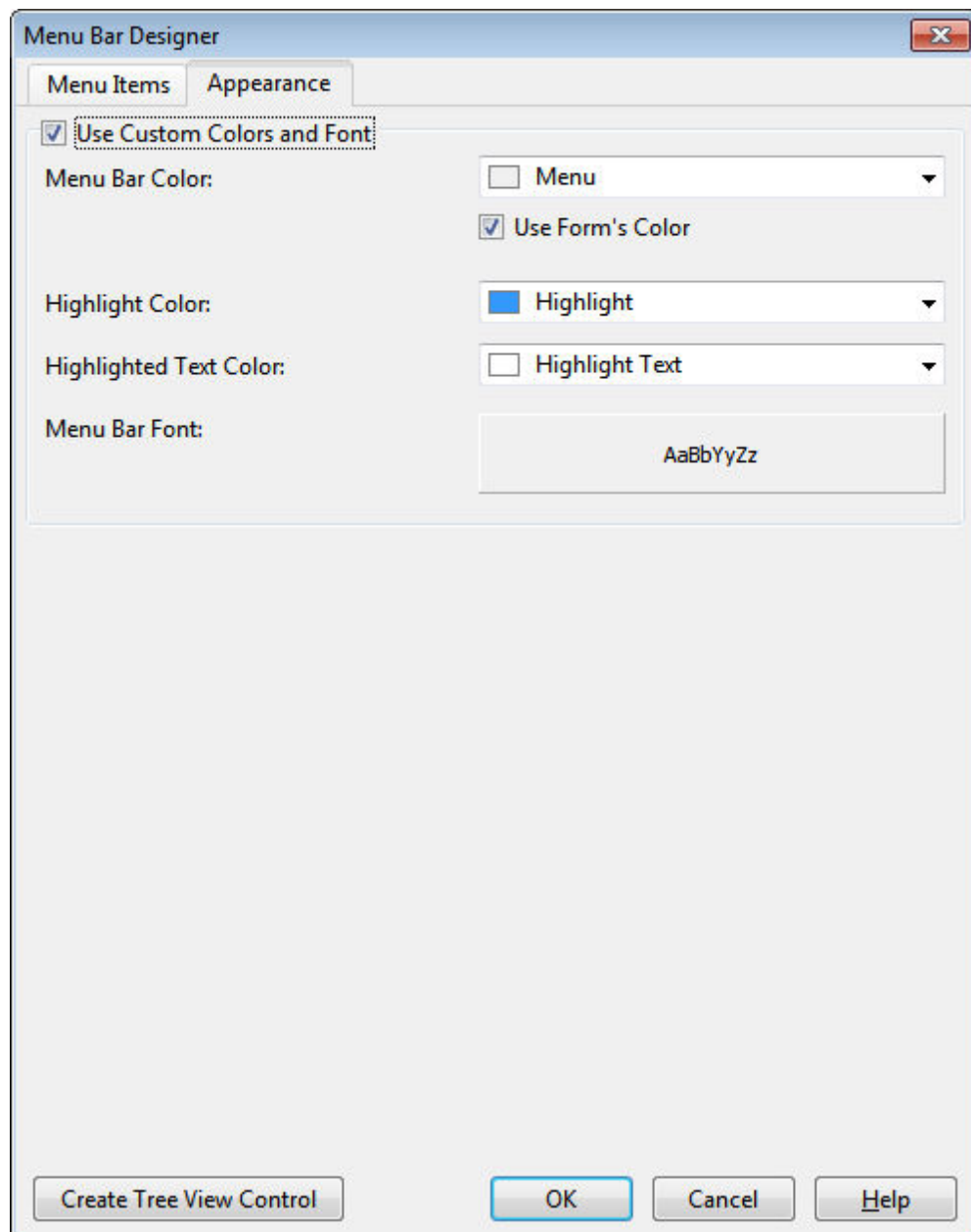
Did you know that now you can also customize the "Appearance" of Menu Bar?

A new "Appearance" tab has been added to further customize the Menu Bar, as follows:

Form Designer > Main Menu > Layout > Design Menu Bar ...

Menu Bar Designer > Appearance

- Use Custom Colors and Font - specifies to use a custom menu bar for the form
- Menu Bar Color - specifies the menu bar color
- Use Forms' Color - specifies to use the form's color for the menu bar
- Highlight Color - specifies the color to be used to draw a highlighted portion of the menu bar
- Highlight Text Color - specifies the text color to be used to when the menu bar is highlighted
- Menu Bar Font - specifies the font for the menu bar



8.30 Customizing the Selected Row Highlight Color in List Views

Did you know that you can customize the Background and Font Color for selected rows in DB and Variable Lookup List Views?

A new "Selected Row Background and Font Color" option for DB Lookup List View and Variable Lookup List View controls is now implemented.

This enhancement will certainly provide a cool visual effect when looking at a custom form with multiple List View controls associated as one-to-many relationship.

Form Designer > DB Lookup List View Control > Object Property > Colors

Selected Row Background and Font Color ...
Background Color:
Font Color:

Form Designer > Variable Lookup List View Control > Object Property > Colors

Selected Row Background and Font Color ...
Background Color:
Font Color:

8.31 Customizing Lookup List View Column Headers/Details

Did you know that now you can customize individual Column Headers and Column Details for Variable/DB Lookup List View?

Two additional options are implemented for Variable Lookup List View and DB Lookup List View Column Properties as follows:

Form Designer > Variable Lookup List View Control > Properties > Columns > Column Properties
 Header Color and Font
 Column Color and Font

Form Designer > DB Lookup List View Control > Properties > Columns > Column Properties
 Header Color and Font
 Column Color and Font

The screenshot shows the 'Variable Lookup List View Properties' dialog box with the 'Columns' tab selected. The 'Columns' list contains four items: 'Column 1 (Company Name)', 'Column 2 (City, State, and Zip Code)', 'Column 3 (Phone Number)', and 'Column 4 (Credit Limit)'. The 'Column Properties' section for 'Column 1' is expanded, showing settings for Caption, Alignment, Min/Max Width, Width, Header Color and Font, Column Color and Font, and Image Field.

Columns

- Column 1 (Company Name)
- Column 2 (City, State, and Zip Code)
- Column 3 (Phone Number)
- Column 4 (Credit Limit)

+ New Item - Delete ↑ ↓

Column Properties

Caption: Company Name

Alignment: Left

Min/Max Width: 246 246

Width: 226 Autosize

Header Color and Font: Custom ... **AaBbYyZz**

Column Color and Font: Custom ... AaBbYyZz

Image Field:

? OK Cancel Apply

The following are examples of using the new options.

Customized Column Headers and Column Details in Variable Lookup List View

	Company Name	City, State, and Zip Code	Phone Number	Credit Limit
1	Barton and Associates	Stony Brook, NY 11794	516-563-4438	\$5,500.00
2	Blue Ridge Technologies, Inc.	Asheville, NC 28801	828-987-6543	\$10,000.00
3	Bytes & Words	Ogden, UT 84403	801-248-4798	\$15,000.00
4	Compdat Computer Consulting	Miami, FL 33128	305-767-4309	\$15,000.00
5	Compumasters Computer Supply	Concord, CA 94521	613-226-7171	\$8,000.00
6	Computer Medical Ctr.	Ft Lauderdale, FL 33309	305-365-7803	\$10,000.00
7	Computer Mountain Inc.,	Denver, CO 80214	303-271-1500	\$10,000.00
8	Computer Warehouse - I	Bloomington, IN 47401	812-701-1002	\$10,000.00
9	Computer Warehouse - II	Boston, CA 02178	617-341-2189	\$10,000.00
10	Data Solutions	Olympia, WA 98504	206-321-5606	\$10,000.00
11	Datacrafters Infosystems	Lynnwood, WA 98037	206-323-4365	\$15,000.00
12	Industrial Computers Inc.	Denver, CO 80214	303-239-7823	\$10,000.00
13	Industrial Concepts Inc.	Livermore, CA 94550	415-878-5600	\$10,000.00
14	Johnson Technologies	Sunnyvale, CA 94089	408-234-8654	\$15,000.00
15	Lanufacturers Discount Computers	Englewood, CO 80112	303-426-8442	\$15,000.00
16	Microcomputer Distribution	Plantation, FL 33317	305-746-2302	\$15,000.00
17	Microtech University - I	Boston, MA 02162	617-423-8921	\$20,000.00
18	Microtech University - II	Appleton, WI 54911	414-245-9867	\$10,000.00
19	Midtown Computer Co.	Chicago, IL 60637	312-277-5000	\$15,000.00
20	MIS by Design	Snoqualmie, WA 99026	509-382-8706	\$6,000.00
21	Modular Software, Inc.	Las Vegas, NV 89104	702-649-9551	\$15,000.00
22	Murrysville Technology Center	Murrysville, PA 15668-1854	724-733-1234	\$100,000.00

Finished

Customized Column Headers and Column Details in Variable Lookup List View

Company Name	City, State, and Zip Code	Phone Number	Credit Limit
1 Barton and Associates	Stony Brook, NY 11794	516-563-4438	\$5,500.00
2 Blue Ridge Technologies, Inc.	Asheville, NC 28801	828-987-6543	\$10,000.00
3 Bytes & Words	Ogden, UT 84403	801-248-4798	\$15,000.00
4 Compdat Computer Consulting	Miami, FL 33128	305-767-4309	\$15,000.00
5 Compumasters Computer Supply	Concord, CA 94521	613-226-7171	\$8,000.00
6 Computer Medical Ctr.	Ft Lauderdale, FL 33309	305-365-7803	\$10,000.00
7 Computer Mountain Inc.,	Denver, CO 80214	303-271-1500	\$10,000.00
8 Computer Warehouse - I	Bloomington, IN 47401	812-701-1002	\$10,000.00
9 Computer Warehouse - II	Boston, CA 02178	617-341-2189	\$10,000.00
10 Data Solutions	Olympia, WA 98504	206-321-5606	\$10,000.00
11 Datacrafters Infosystems	Lynnwood, WA 98037	206-323-4365	\$15,000.00
12 Industrial Computers Inc.	Denver, CO 80214	303-239-7823	\$10,000.00
13 Industrial Concepts Inc.	Livermore, CA 94550	415-878-5600	\$10,000.00
14 Johnson Technologies	Sunnyvale, CA 94089	408-234-8654	\$15,000.00
15 Lanufacturers Discount Computers	Englewood, CO 80112	303-426-8442	\$15,000.00
16 Microcomputer Distribution	Plantation, FL 33317	305-746-2302	\$15,000.00
17 Microtech University - I	Boston, MA 02162	617-423-8921	\$20,000.00
18 Microtech University - II	Appleton, WI 54911	414-245-9867	\$10,000.00
19 Midtown Computer Co.	Chicago, IL 60637	312-277-5000	\$15,000.00
20 MIS by Design	Snoqualmie, WA 99026	509-382-8706	\$6,000.00
21 Modular Software, Inc.	Las Vegas, NV 89104	702-649-9551	\$15,000.00
22 Murrysville Technology Center	Murrysville, PA 15668-1854	724-733-1234	\$100,000.00

Finished

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

8.32 Defining Custom EEPs for "On Roll Up" and "On Roll Down"

R:BASE includes a cool option of collapsible roll-up forms, a form that will reduce its height to nothing but the title bar when clicked.

To enable the roll up option from within the Form Designer:

1. Open a form's "Form Properties"
2. Select the "Dynamic Caption" tab
3. Enable the "Roll Up Button" check box

By checking this option, you may create a roll-down and roll-up effect for a form when a user right-clicks its title bar button.

In addition, now you can define a Form Custom EEP "On Roll Up" and "On Roll Down" to control anything you can imagine.

Form Designer | Form Properties | EEPs ...

Notice the two additional EEP options for:

- On Roll Up
- On Roll Down

8.33 Defining TEMPORARY TABLES/VIEWS with Actions

Did you know that now you can define all Form, Label or Report related TEMPORARY TABLES/VIEWS in "On Before Design Action..."?

The latest R:BASE supports the option to define Form, Label or Report Master and Slave tables or views based on TEMPORARY tables or view before the form, label or report is brought up in the designer.

This enhancement will eliminate the most common mistake when opening the form, label or report based on temporary table/view, if not pre-defined at the R> prompt or in a startup command file.

For example, if you open the "SalesOrder" form in RRBW19 with pre-defining the required temporary tables (tInvoiceHeader and tInvoiceDetail), you'll notice the "-ERROR- Table tInvoiceHeader Does Not Exist!" and you are not able to open the form until you RUN TempTables.RMD to define all temporary tables/views required for forms/labels/reports.

Well, now all you have to do is include the following block of code in SalesOrder form as "On Before Design Action...".

Here's how:

Form Designer | Layout | On Before Design Action...

```

SET ERROR MESSAGE 2038 OFF
DROP TABLE tInvoiceHeader
CREATE TEMPORARY TABLE `tInvoiceHeader` +
(`TransID` INTEGER, +
`CustID` INTEGER, +
`EmpID` INTEGER, +
`TransDate` DATE, +
`BillToCompany` TEXT (40), +
`BillToAddress` TEXT (30), +
`BillToCity` TEXT (20), +
`BillToState` TEXT (2), +
`BillToZip` TEXT (10), +
`ShipToCompany` TEXT (40), +
`ShipToAddress` TEXT (30), +
`ShipToCity` TEXT (20), +
`ShipToState` TEXT (2), +
`ShipToZip` TEXT (10), +
`NetAmount` CURRENCY, +
`Freight` = ( netamount* .01) CURRENCY, +
`Tax` = ( netamount* .081) CURRENCY, +
`InvoiceTotal` = (NetAmount+Freight+Tax) CURRENCY)
COMMENT ON TABLE tInvoiceHeader IS 'Invoice Header Information'
DROP TABLE tInvoiceDetail
CREATE TEMPORARY TABLE `tInvoiceDetail` +
(`TransID` INTEGER, +
`DetailNum` INTEGER, +
`Model` TEXT (6), +
`Units` INTEGER, +
`Price` CURRENCY, +
`Discount` REAL, +
`SalePrice` = (Price-(Price*Discount/100)) CURRENCY, +
`ExtPrice` = (Units* SalePrice) CURRENCY)
COMMENT ON TABLE tInvoiceDetail IS 'Invoice Detail Information'
SET ERROR MESSAGE 2038 ON

```

RETURN

Save the form and close the form designer.

If you disconnect the database (when all temporary tables/view are automatically dropped), you'll never have to worry about opening that form in the form designer next time. The required TEMPORARY tables will automatically be defined before opening the form in form designer.

These features also allow your forms, labels and reports to be totally portable modules for your ever-changing applications.

8.34 Defining the Number of Items in Pop-up Menus

Did you know that you can define more than 18 items/lines to be displayed for Pop-up Menu used in DB Edit or Variable Edit Controls?

Using the latest versions/updates, you are no longer limited to 18 items/lines. You may even pick your own number, if you wish.

Previous versions are restricted to 18 items/lines for Pop-up Menu.

Form Designer > DB Edit Control > Object Property > Pop-up Menu
Form Designer > Variable Edit Control > Object Property > Po-up Menu

8.35 Designing Custom Pop-up Menu

Did you know that now you can customize all Pop-up Menus for DB Edit and Variable Edit Controls to look like the ListBox with Show Lines, Title Background Color, Title Font/Color, List Background Color, and List Font/Color?

To customize the DB Edit and Variable Edit Pop-up Menus:

Form Designer > DB Edit > DB Edit Properties > Pop-up Menu

or

Form Designer > DB Edit > Variable Edit Properties > Pop-up Menu

Notice the following additional options:

- Show Lines (to separate multiple columns for a better look)
- Title Font and Color
- List Font and Color
- Title Background Color
- List Background Color

8.36 Disabling the Sort Feature for DB Grid Columns

Did you know that you can disable the column sort option used in DB Grid?

When a DB Grid control is placed on a form, by default, every column includes the ability to sort (Asc/Des) when clicking on the column title; displaying the current sort state. However, there may be application instances, where you may not want your users to change the default sort state.

A new option [] Title Clickable has been added for customizing the individual column properties for DB Grids. The default is set to checked, i.e, Title Clickable.

A new PROPERTY command is also introduced to manage the column title properties:

Syntax:

```
PROPERTY <DBGRIDComponentID> Columns[n]-> ClickableTitle 'value'
```

Example:

```
PROPERTY DBGRID Columns[0]-> ClickableTitle 'FALSE'
```

This property command will disable the sort feature for the first DB Grid column.

8.37 Displaying a Balloon Tip in DB/Variable Edit Fields

Did you know that you can display a balloon tip in DB Edit and Variable Edit Fields?

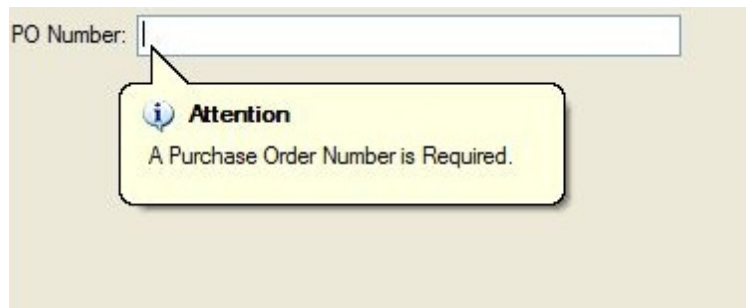
The balloon tip displays a hint message that can be used in response to an end user action on the form, or even to a specific form object. The balloon tip will appear when the SHOWBALLOONTIP parameter is set to TRUE for the field's Component ID. A title, message, and icon image can be displayed in the balloon tip. The balloon tip is only available to DB Edit and Variable Edit fields.

Icon options available include:

- NONE
- INFO
- WARNING
- ERROR
- INFO_LARGE (Windows 7 and higher only)
- WARNING_LARGE (Windows 7 and higher only)
- ERROR_LARGE (Windows 7 and higher only)

Example: (Variable Edit On Exit EEP)

```
IF vPONumber IS NULL THEN
  PROPERTY Pedit_PO ShowBalloonTip TRUE
  PROPERTY Pedit_PO BalloonTipTitle 'Attention'
  PROPERTY Pedit_PO BalloonTipText 'A Purchase Order Number is Required.'
  PROPERTY Pedit_PO BalloonTipIcon INFO
  PROPERTY Pedit_PO SET_FOCUS 'TRUE'
ENDIF
RETURN
```



8.38 Displaying a Cue Banner in DB/Variable Edit Fields

Did you know that you can display cue banner in DB Edit and Variable Edit Fields?

Cue banners (aka watermarks) are text hints that can be displayed as a prompt to the user inside an DB Edit or Variable Edit control. This is a line of "grayed out" text which can be used to prompt the user for the information s/he is expected to enter in that box. For example if the edit control receives a search string, the cue banner may read "Enter search text here". When the user clicks in the box, the text goes away and s/he can type entry as usual.

Once the focus lands on the field, or when keystrokes are entered, the cue banner disappears. An optional setting can be used so the hint remains until keystrokes are pressed (Windows 7 and higher only).

The PROPERTY command can also be used to specify the cue banner for the DB/Variable Edit fields.

Example:

```
-- Form On After Start Custom EEP  
PROPERTY DB_UserName CUEBANNER 'User Name'  
PROPERTY DB_Password CUEBANNER 'Password'  
RETURN
```



The screenshot shows a 'Login' dialog box with a light blue title bar. It contains two text input fields. The first field is labeled 'User Name:' and contains the text 'Enter User Name'. The second field is labeled 'Password:' and contains the text 'Enter Password'. At the bottom of the dialog, there are three buttons: a help button (question mark icon), a 'Login' button (key icon), and a 'Cancel' button (folder icon).



The screenshot shows the same 'Login' dialog box. The 'User Name:' field is now highlighted in yellow, indicating it has focus. The 'Password:' field still contains the text 'Enter Password'. The buttons at the bottom remain the same.

8.39 Displaying a Row Number in List Views

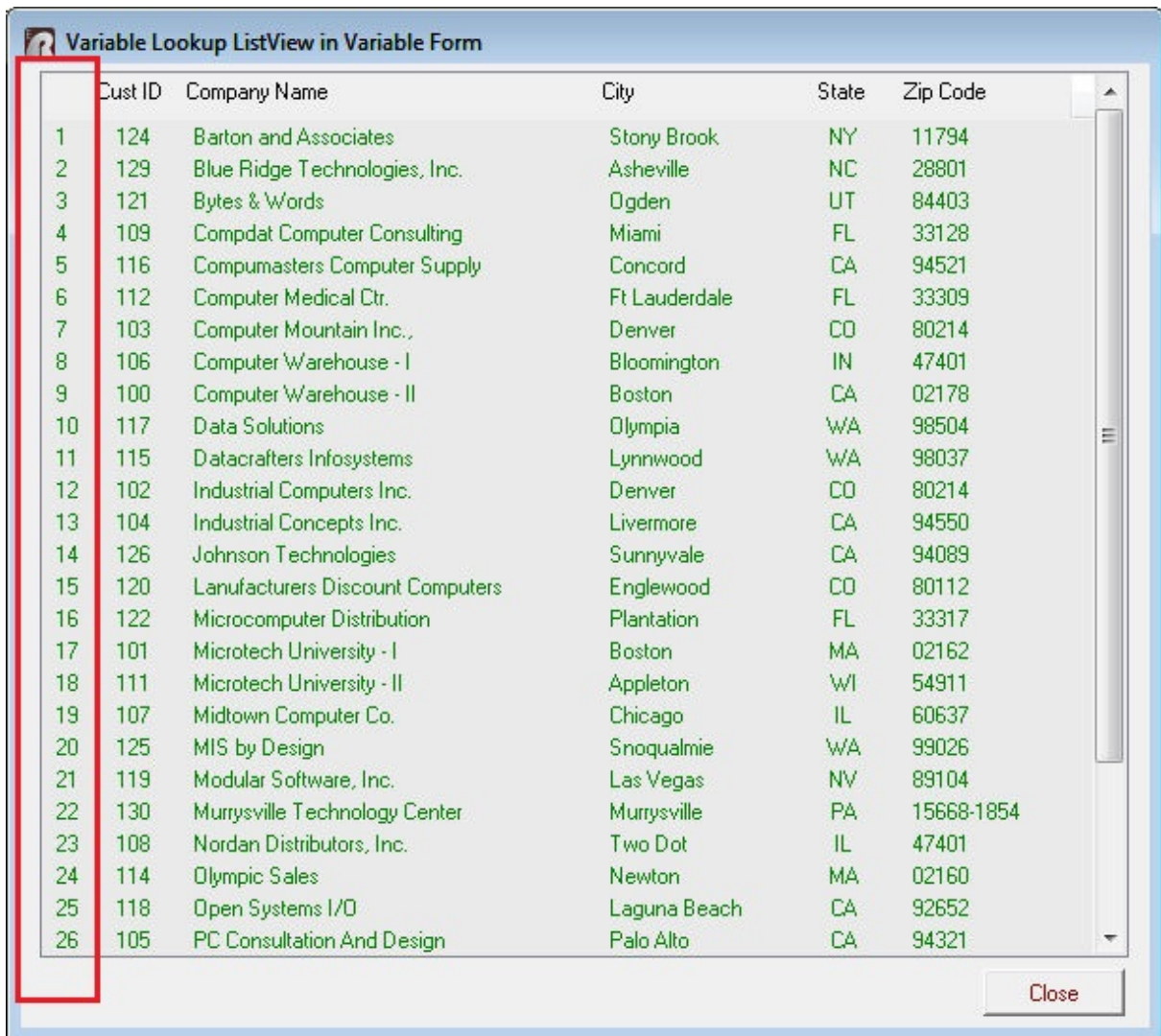
Did you know that you can display Row Number in List View Control?

Here's how ...

1. Start R:BASE
2. While in Form Designer, right-click on the Variable Lookup List View control and check the option for [] Show Row Numbers and click on [Apply] button.

Variable Lookup List View Properties | Effects | Show Row Numbers

3. Now run the form or use [Ctrl + F9] to preview the form



	Cust ID	Company Name	City	State	Zip Code
1	124	Barton and Associates	Stony Brook	NY	11794
2	129	Blue Ridge Technologies, Inc.	Asheville	NC	28801
3	121	Bytes & Words	Ogden	UT	84403
4	109	Compdat Computer Consulting	Miami	FL	33128
5	116	Compumasters Computer Supply	Concord	CA	94521
6	112	Computer Medical Ctr.	Ft Lauderdale	FL	33309
7	103	Computer Mountain Inc.,	Denver	CO	80214
8	106	Computer Warehouse - I	Bloomington	IN	47401
9	100	Computer Warehouse - II	Boston	CA	02178
10	117	Data Solutions	Olympia	WA	98504
11	115	Datacrafters Infosystems	Lynnwood	WA	98037
12	102	Industrial Computers Inc.	Denver	CO	80214
13	104	Industrial Concepts Inc.	Livermore	CA	94550
14	126	Johnson Technologies	Sunnyvale	CA	94089
15	120	Lanufacturers Discount Computers	Englewood	CO	80112
16	122	Microcomputer Distribution	Plantation	FL	33317
17	101	Microtech University - I	Boston	MA	02162
18	111	Microtech University - II	Appleton	WI	54911
19	107	Midtown Computer Co.	Chicago	IL	60637
20	125	MIS by Design	Snoqualmie	WA	99026
21	119	Modular Software, Inc.	Las Vegas	NV	89104
22	130	Murrysville Technology Center	Murrysville	PA	15668-1854
23	108	Nordan Distributors, Inc.	Two Dot	IL	47401
24	114	Olympic Sales	Newton	MA	02160
25	118	Open Systems I/O	Laguna Beach	CA	92652
26	105	PC Consultation And Design	Palo Alto	CA	94321

Similar option is also available for DB Lookup List View control.

You may also dynamically add Row Number column using the following

PROPERTY Command:

```
PROPERTY ListViewControlCompID SHOWROWNUMBERS 'TRUE'
```


Notes:

1. New Row Number column will be the first column on left.
2. Adding Row Number column will offset the defined width of List View Control.
3. Depending on the number of rows to be displayed, you will have to expand the width of List View Control accordingly.

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

8.40 Displaying the Current Table/Field on the Status Bar

Did you know that you can display the Current Table and Current Field on the form Status Bar?

After placing the Status Bar control, you may add one of the following Status Bar Panes, including "Current Table" and "Current Field":

- Clock Status
- Key Status
- Simple Status
- Glyph Status
- Marquee Status
- Table State Status
- DB Field Status
- Current Table
- Current Field

8.41 Displaying the Record Number in a DB Grid

Did you know that you can display the record number in DB Grid?

Here's how:

1. Form Designer > DB Grid Control > Properties > DB Grid Options
2. [x] Check the option for "Record Number" and select the "OK" button
3. Save and close the Form Designer
4. Database Explorer > Select Form > Run Form

The left column will display the actual record number for the table/view.

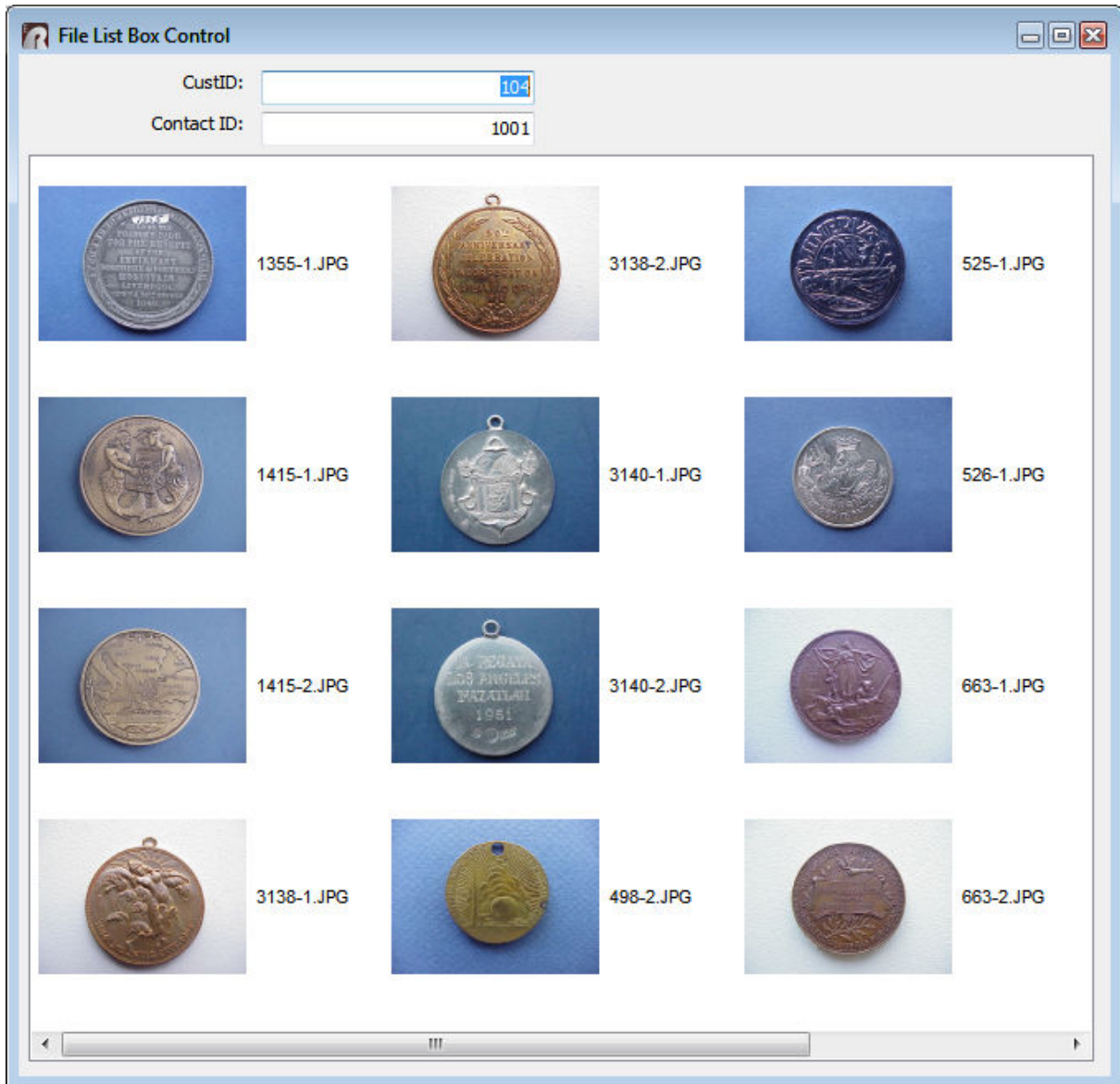
8.42 Displaying Thumbnail Images With the File List Box

Did you know that you can display file thumbnails within the File List Box control?

A new "Extract Thumbnail Cache" setting allows for the control to display images for files within a folder. The setting gets the file's thumbnail, which is not limited to image files.

It can also extract thumbnails of other files, just like Windows Explorer. To increase the size of the thumbnail, adjust the "Item Height" setting, to perhaps 128.

The following is an example of what the File List Box can display.



Supported Versions:

- R:BASE X (Build...: 10.0.2.11227 or higher)

8.43 Documenting Form EEPs

Did you know that while in Form Designer, you can document all Embedded Custom EPPs (if defined), as well as External EEPs (if referenced)?

In addition to documenting Custom EEPs (if defined), now you may also document all External EEPs (if referenced). This will help documenting all references related to Form External EEPs when converting legacy databases and applications.

Here's how:

While in Form Designer, select "Form" > "Document Custom EEPs" from the main Menu Bar.

Four output options are available:

- Printer
- Text File
- PDF File
- Clipboard
- **In Text Editor**
- **EEP Map**

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

8.44 Dynamically Changing Band List Properties

Did you know that you can dynamically change the properties of Enhanced DB Grid Band List(s), including the CAPTION and FONT.

Now you may use the following GETPROPERTY/PROPERTY commands:

- To Obtain Band Count in Enhanced DB Grid
GETPROPERTY ComponentID BANDS->COUNT vValue
- To obtain Current Caption for a Given Band in Enhanced DB Grid
GETPROPERTY ComponentID BANDS[n]->CAPTION vValue
- To Dynamically Change Caption for a Given Band in Enhanced DB Grid
PROPERTY ComponentID BANDS[n]->CAPTION value
- To Dynamically Change Font Name for Band(s) in Enhanced DB Grid
PROPERTY ComponentID BANDS->FONT_NAME value
- To Change Font Color for Band(s) in Enhanced DB Grid
PROPERTY ComponentID BANDS->FONT_COLOR Value
- To Dynamically Change Font Size for Band(s) in Enhanced DB Grid
PROPERTY ComponentID BANDS->FONT_SIZE value
- To Dynamically Change Font Height for Band(s) in Enhanced DB Grid
PROPERTY ComponentID BANDS->FONT_HEIGHT value
- To Dynamically Change Font Bold Style for Band(s) in Enhanced DB Grid
PROPERTY ComponentID BANDS->FONT_BOLD value
Values TRUE/FALSE
- To Dynamically Change Font Italic Style for Band(s) in Enhanced DB Grid
PROPERTY ComponentID BANDS->FONT_ITALIC value
Values TRUE/FALSE
- To Dynamically Change Font Strikeout Style for Band(s) in Enhanced DB Grid
PROPERTY ComponentID BANDS->FONT_STRIKEOUT value
Values TRUE/FALSE
- To Dynamically Change Font Underline Style for Band(s) in Enhanced DB Grid
PROPERTY ComponentID BANDS->FONT_UNDERLINE value
Values TRUE/FALSE
- Example 01
-- Database: RRBYW19
-- Form Name: EnhancedDBGrid
-- Control: Enhanced DB Grid
-- Related Tables: Customers, InvoiceHeader

```
-- Component ID: DBGrid_Customers, DBGrid_Invoices
-- Band List Count: 1
-- Bit Button: [Change Band Properties]
-- Custom EEP:
PROPERTY DBGrid_Invoices BANDS[0]->CAPTION 'Selected Customer Invoice
Details'
PROPERTY DBGrid_Invoices BANDS->FONT_NAME Tahoma
PROPERTY DBGrid_Invoices BANDS->FONT_COLOR GREEN
PROPERTY DBGrid_Invoices BANDS->FONT_SIZE 8
PROPERTY DBGrid_Invoices BANDS->FONT_BOLD TRUE
RETURN
```

Enhanced DB Grid Band List Before:

The screenshot displays a software application window with two main data grids and a control panel on the right.

Customers Grid:

State	Company	City	Zip Code	Status	Web Site
CA	Computer Warehouse - II	Boston	02178	<input checked="" type="checkbox"/>	http://www.cwi.com
MA	Microtech University - I	Boston	02162	<input checked="" type="checkbox"/>	http://www.mui.com
CO	Industrial Computers Inc.	Denver	80214	<input checked="" type="checkbox"/>	http://www.ici.com
CO	Computer Mountain Inc.,	Denver	80214	<input checked="" type="checkbox"/>	http://www.cmi.com
CA	Industrial Concepts Inc.	Livermore	94550	<input checked="" type="checkbox"/>	http://www.ici.com
CA	PC Consultation And Design	Palo Alto	94321	<input checked="" type="checkbox"/>	http://www.pccad.com
IN	Computer Warehouse - I	Bloomington	47401	<input checked="" type="checkbox"/>	http://www.cwii.com
IL	Midtown Computer Co.	Chicago	60637	<input checked="" type="checkbox"/>	http://www.mco.com
IL	Nordan Distributors, Inc.	Two Dot	47401	<input checked="" type="checkbox"/>	http://www.mdi.com
FL	Compdat Computer Consulting	Miami	33128	<input checked="" type="checkbox"/>	http://www.ccc.com
WA	Softech Database Design	Mercer Island	98040	<input checked="" type="checkbox"/>	http://www.sdd.com

Invoices Grid:

Invoice No.	Date	Invoice Details			
		Net Amount	Freight	Tax	Invoice Total
1009	03/21/2011	\$7,392.00	\$73.92	\$598.75	\$8,064.67
1079	12/23/2011	\$26,559.00	\$265.59	\$2,151.28	\$28,975.87
		\$33,951.00	\$339.51	\$2,750.03	\$37,040.54

Right Panel:

- Navigation buttons: First, Prev, Next, Last, Close.
- Selected Invoice: 1079**
- Print Invoice button.
- Invoice Summary button.

Enhanced DB Grid Band List **After**:

Customers					
State	Company	City	Zip Code	Status	Web Site
CA	Computer Warehouse - II	Boston	02178	<input checked="" type="checkbox"/>	http://www.cwI.com
MA	Microtech University - I	Boston	02162	<input checked="" type="checkbox"/>	http://www.mui.com
CO	Industrial Computers Inc.	Denver	80214	<input checked="" type="checkbox"/>	http://www.ici.com
CO	Computer Mountain Inc.,	Denver	80214	<input checked="" type="checkbox"/>	http://www.cmi.com
CA	Industrial Concepts Inc.	Livermore	94550	<input checked="" type="checkbox"/>	http://www.ici.com
CA	PC Consultation And Design	Palo Alto	94321	<input checked="" type="checkbox"/>	http://www.pccad.com
IN	Computer Warehouse - I	Bloomington	47401	<input checked="" type="checkbox"/>	http://www.cwii.com
IL	Midtown Computer Co.	Chicago	60637	<input checked="" type="checkbox"/>	http://www.mco.com
IL	Nordan Distributors, Inc.	Two Dot	47401	<input checked="" type="checkbox"/>	http://www.mdi.com
FL	Compdat Computer Consulting	Miami	33128	<input checked="" type="checkbox"/>	http://www.ccc.com
WA	Softtech Database Design	Mercer Island	98040	<input checked="" type="checkbox"/>	http://www.sdd.com

Invoices					
Invoice No.	Date	Selected Customer Invoice Details			
		Net Amount	Freight	Tax	InvoiceTotal
1009	03/21/2011	\$7,392.00	\$73.92	\$598.75	\$8,064.67
1079	12/23/2011	\$26,559.00	\$265.59	\$2,151.28	\$28,975.87
		\$33,951.00	\$339.51	\$2,750.03	\$37,040.54

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

8.45 Dynamically Passing DB Grid Filter Parameters

In addition to many New and Enhanced Controls and Properties in R:BASE, Enhanced DB Grid in Form Designer includes the option to "Show Filter Bar".

Enabling this option will display a filter bar directly under the titles which allows end-user to enter value to limit the displayed grid records based on matching criteria. You may also pre-define the Filter Pick List by selecting "Column Values", if you wish.

Did you know that you can dynamically pass filter parameters as "On After Start Custom EEP" or "On Demand", using the expanded Enhanced DB Grid Filter PROPERTY commands?

Here is a comprehensive list of supported PROPERTY/GETPROPERTY commands that are specific to Enhanced DB Grid Filter Bar.

```
-- Passing Filter Value for a Given Column Number
-- Syntax
PROPERTY CompID FILTERS[NNN]->VALUE filter_value
-- Example
PROPERTY EDBGridCust FILTERS[2]->VALUE 'CA'

-- Passing Filter Value for Given Column Name
-- Syntax
```

```
PROPERTY CompID FILTERS->COLUMN_NAME->ColName filter_value
-- Example
PROPERTY EDBGridCust FILTERS->COLUMN_NAME->CustState 'CA'

-- Capturing Filter Value for Given Column Number
-- Syntax
GETPROPERTY CompID FILTERS[NNN]->VALUE VarName
-- Example:
GETPROPERTY EDBGridCust FILTERS[2]->VALUE vStateFilter

-- Capturing Filter Value for Given Column Name
-- Syntax
GETPROPERTY CompID FILTERS->COLUMN_NAME->ColName VarName
-- Example
GETPROPERTY EDBGridCust FILTERS->COLUMN_NAME->CustState vValue

-- Clearing Filter Value for Given Column Number
-- Syntax
PROPERTY CompID FILTERS[NNN]->CLEAR_ITEMS TRUE
-- Example
PROPERTY EDBGridCust FILTERS[2]->CLEAR_ITEMS TRUE

-- Clearing Filter Value for Given Column Name
-- Syntax
PROPERTY CompID FILTERS->COLUMN_NAME->ColName->CLEAR_ITEMS TRUE
-- Example
PROPERTY EDBGridCust FILTERS->COLUMN_NAME->CustState->CLEAR_ITEMS TRUE

-- Dynamically Defining Single Filter Item for a Given Column Number
-- Syntax
PROPERTY ComponentID FILTERS[NNN]->ADD_ITEMS Value
-- Example
PROPERTY EnhDBGridCust FILTERS[2]->ADD_ITEMS 'MO'

-- Dynamically Defining Single Filter Item for a Given Column Name
-- Syntax
PROPERTY CompID FILTERS->COLUMN_NAME->ColName->ADD_ITEM ItemValue
-- Example
PROPERTY EDBGridCust FILTERS->COLUMN_NAME->CustState->ADD_ITEMS 'MO'

-- Dynamically Defining Multiple Filter Items for a Given Column Number
-- Syntax
PROPERTY CompID FILTERS[NNN]->ADD_ITEMS Value
-- Example:
PROPERTY EDBGridCust FILTERS[2]->ADD_ITEMS 'MO|HN|NY|CT'

-- Dynamically Defining Multiple Filter Items for a Given Column Name
-- Syntax
PROPERTY CompID FILTERS->COLUMN_NAME->ColName->ADD_ITEMS Value
-- Example
PROPERTY EDBGridCust FILTERS->COLUMN_NAME->CustState->ADD_ITEMS 'MO|HN|NY|
CT'
```

```

-- To repopulate the filter from column values for a given column number
-- Syntax
PROPERTY CompID FILTERS[NNN]->ADD_COLUMN_VALUES TRUE
-- Example
PROPERTY EDBGridCust FILTERS[2]->ADD_COLUMN_VALUES TRUE

-- To repopulate the filter from column values for a given column name
-- Syntax
PROPERTY CompID FILTERS->COLUMN_NAME->ColName->ADD_COLUMN_VALUES TRUE
-- Example
PROPERTY EDBGridCust FILTERS->COLUMN_NAME->CustState->ADD_COLUMN_VALUES
TRUE

-- To clear all filter values at once
-- Syntax
PROPERTY CompID CLEARFILTER TRUE
-- Example
PROPERTY EDBGridCust CLEARFILTER TRUE

```

8.46 Easily Saving Table Image Data as a File

Did you know you can easily save images stored in R:BASE tables or variable objects to an external file?

A PROPERTY command parameter has been added to save an image to a file from the DB Image, Variable Image, and Image form controls.

Syntax:

```
PROPERTY ComponentID SaveImageToFile vFileName
```

The SaveImageToFile parameter saves the displayed image from the form image control to the specified image file and path. The ComponentID value references the specific image control to save the image from.

Supported Versions:

- R:BASE X (Build...: 10.0.2.11130 or higher)

8.47 Enabling/Disabling the "Drag and Drop" Feature

Did you know that you can enable or disable the "drag & drop" feature of any form control using the PROPERTY command?

A new DRAGDROPENABLED parameter has been added to the PROPERTY command to enable or disable the "drag & drop" feature of any form control, on demand.

Syntax:

```
PROPERTY <FormControlCompID> DRAGDROPENABLED 'value'
Values: (TRUE/FALSE)
```

-- Example 01 (To enable Drag and Drop feature):

```
PROPERTY <FormControlCompID> DRAGDROPENABLED 'TRUE'
RETURN
```

-- Example 02 (To disable Drag and Drop feature):

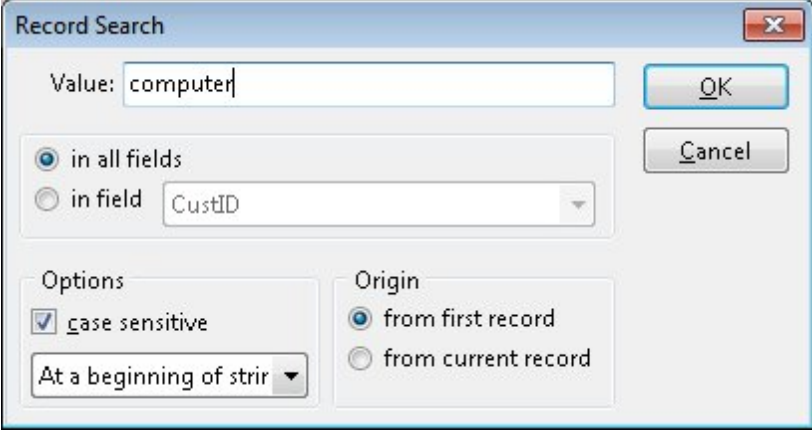
```
PROPERTY <FormControlCompID> DRAGDROPENABLED 'FALSE'
```

RETURN

8.48 Enhanced DB Grid Control - Hot Keys

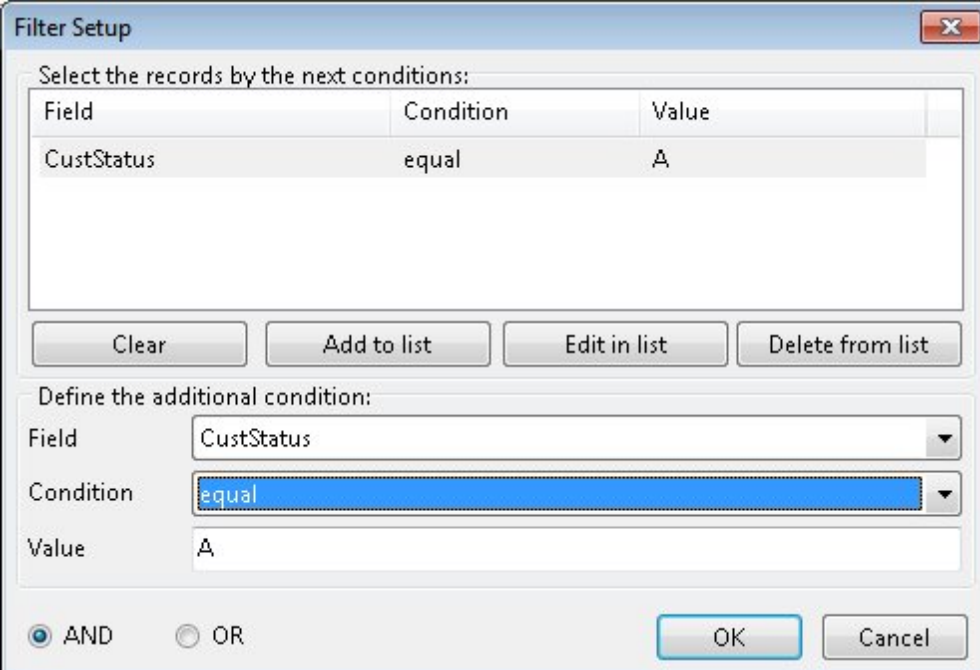
In addition to all enhanced features of Enhanced DB Grid Control in Forms, you may also use the following hot keys to extend the use of Enhanced DB Grid Control.

01. Enhanced DB Grid: Use [Ctrl + F3] to Search Records



The Record Search dialog box is used to search for records in a database grid. It features a text input field for the search value, which currently contains "computer". Below this, there are two radio buttons: "in all fields" (selected) and "in field" (with a dropdown menu showing "CustID"). The "Options" section includes a checked checkbox for "case sensitive" and a dropdown menu set to "At a beginning of strir". The "Origin" section has two radio buttons: "from first record" (selected) and "from current record". "OK" and "Cancel" buttons are located on the right side.

02. Enhanced DB Grid: Use [Ctrl + L] to Customize Filters

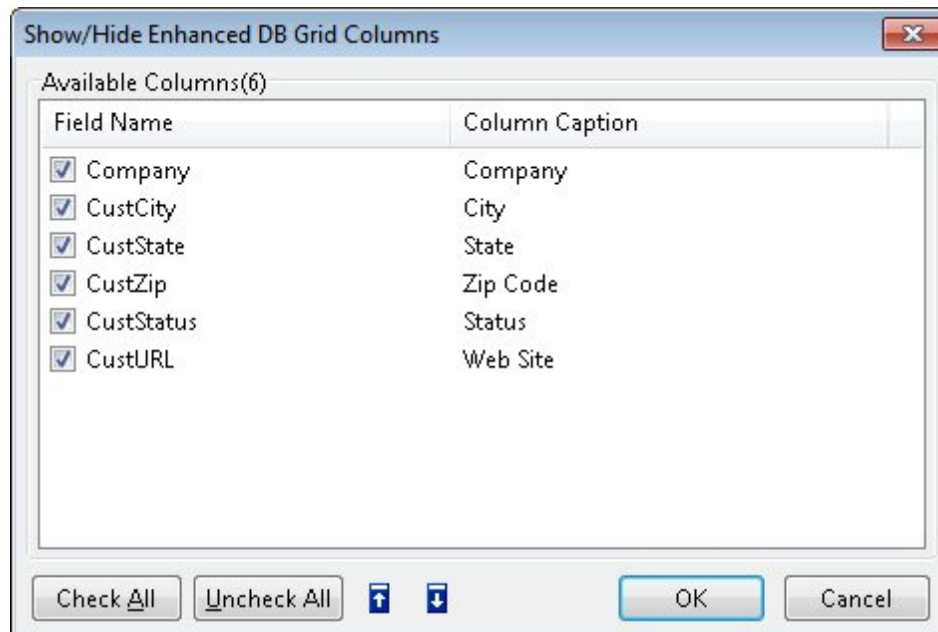


The Filter Setup dialog box allows users to define search filters. It contains a table with the following data:

Field	Condition	Value
CustStatus	equal	A

Below the table are buttons for "Clear", "Add to list", "Edit in list", and "Delete from list". The "Define the additional condition:" section has a "Field" dropdown set to "CustStatus", a "Condition" dropdown set to "equal", and a "Value" text field containing "A". At the bottom, there are radio buttons for "AND" (selected) and "OR", along with "OK" and "Cancel" buttons.

03. Enhanced DB Grid: Use [Ctrl + S] to Show/Hide Grid Columns



You may also use the same hot keys to re-arrange columns.

8.49 Focusing on the Enhanced DB Grid Filter Bar

When deploying Enhanced DB Grids on Forms, you might wish to place the cursor directly onto the "Filter" as soon as the form opens.

Did you know that now you can use a new "FocusOnFilterBar" PROPERTY command parameter to set the focus/cursor directly in a filter field for an Enhanced DB Grid?

Here's how:

```
PROPERTY EBDGrid_ComponentID FocusOnFilterBar 'TRUE'
```

Using this approach, you can have more control over your Enhanced DB Grid driven forms.

8.50 Getting the Radio Button Count

Did you know that you can get the count of total number of radio buttons in a RadioGroup using the GETPROPERTY command?

Here's how:

```
GETPROPERTY <RadioGroupCompID> 'BUTTONCOUNT' varname
SET VAR vButtonCount = (INT(.varname))
RETURN
```

Variable varname will return the text value of a total number of Radio Buttons in DB/Variable RadioGroup control.

Variable vButtonCount will return the integer value of button count (varname).

8.51 GROUPINDEX property for Speed Buttons

Did you know that you can group all Speed Buttons to determine how the buttons behave when clicked?

You can use the new PROPERTY command GROUPINDEX parameter for Speed Buttons to allow all speed buttons to work together as a group.

Syntax:

```
PROPERTY <SpeedButtonComponentID> GROUPINDEX nnnn
```

You will need to set GROUPINDEX to determine how the button behaves when clicked.

-- Example:

```
PROPERTY <SpeedButtonOFF> GROUPINDEX 1000
```

```
PROPERTY <SpeedButtonON> GROUPINDEX 1000
```

```
PROPERTY <SpeedButtonOFF> DOWN 'TRUE'
```

```
PROPERTY <SpeedButtonON> DOWN 'FALSE'
```

When GROUPINDEX is 0, the button behaves independently of all other buttons on the form. When the user clicks such a speed button, the button appears pressed (in its clicked state) and then returns to its normal up state when the user releases the mouse button.

When GROUPINDEX is greater than 0, the button remains selected (in its down state) when clicked by the user. When the user clicks a selected button, it returns to the up state, unless Caption is False. Setting the GROUPINDEX property of a single speed button to a value greater than 0 causes the button to behave as a two-state button when Caption is True.

Speed buttons with the same GROUPINDEX property value (other than 0), work together as a group. When the user clicks one of these buttons, it remains selected until the user clicks another speed button belonging to the same group. Speed buttons used in this way can present mutually exclusive choices to the user.

The option DOWN specifies whether the button is selected (down) or unselected (up). Set DOWN to determine whether a speed button is selected. The DOWN property only applies if the GROUPINDEX property of the button is nonzero.

When GROUPINDEX is greater than 0, set DOWN to TRUE to select a button. When the user clicks on a button in the unselected (up) state, the button is selected and DOWN is set to TRUE. When the user clicks on a button in the selected (down) state, if Caption is True, the button becomes unselected and DOWN is set to FALSE.

At design time, specify which button in a group is the initially selected button by setting the Down property of the selected button to TRUE.

Note: When GROUPINDEX is 0, buttons do not remain in the selected state when clicked.

8.52 Listing All Controls with Component IDs

Did you know that you can list all controls with Component IDs, while in a form designer? Here's how:

While in Form Designer, press the [F3] key to launch the Data Dictionary.

Notice the additional tabs...

Now click on the [Form Component IDs] tab

The list will display all Component ID-associated form controls and the control caption (if exists).

While in Form Designer, you may also list "Form Actions", if defined, by clicking on the [Form Actions] tab of data dictionary.

8.53 Loading Images in Drop-Down and Bit Buttons

Did you know that you can load images in Drop-Down and Bit Buttons from an executable (.EXE), Dynamic Link Library (.DLL), OCX file, or icon (.ICO) file?

The Drop Down Menu Button and Bit Button permit the ability to load a glyph from an executable (.exe), dynamic link library (.dll), OCX file, or icon (.ico) file. A 32x32 or 16x16 pixel glyph can be displayed.

In addition, the Drop Down Menu Button and Bit Button now provide alpha channel support to display a greater level of transparency for glyphs used on the buttons. A new attribute "Blend Image" allows this capability. The highest benefit of using alpha channel is visual appeal. Without alpha channel support, images would have a blocky jagged edge to them, like displayed below for the button on the left. This allows for the buttons to support PNG images in addition to BMP.

8.54 Making an Enhanced Group Box Transparent

When using "Enhanced GroupBox" Control on the form, if you would like the Enhanced GroupBox to acquire the background color and/or application theme of your form wallpaper, etc., try the following:

1. Start R:BASE
2. CONNECT dbname
3. Database Explorer > Forms
 - Highlight the form with Enhanced GroupBox
 - Right-Click > choose "Design Form"
4. While in Form Designer:
 - Right-Click on Enhanced GroupBox Object placed on the form
 - Select Object Property
 - [x] Transparent (set to True)
 - Click on "Effects" Tab
 - Select Border: "Group Style" as "Custom"
 - Click on [OK] button to save property.
5. Click on "Run Form" button or use [Ctrl+F9] to Preview
 - Close the form

8.55 Managing Digital Clock Properties

Did you know that you can change the default 24-hour Digital Clock control in forms to display a 12-hour (AM/PM) clock by using the TWELVEHOURCLOCK parameter in the PROPERTY command?

Syntax:

```
PROPERTY <DigitalClockCompID> TWELVEHOURCLOCK 'value'
```

The default is set to 'FALSE'.

-- Example 01: To Display 12 Hour (AM/PM) Clock

```
PROPERTY DigitalClockCompID TWELVEHOURCLOCK 'TRUE'
```

```
-- Example 02: To Display 24 Hour Clock
```

```
PROPERTY DigitalClockCompID TWELVEHOURCLOCK 'FALSE'
```

In addition, you may also use the following PROPERTY command enable/disable the display of "Seconds" in Digital Clock.

Syntax:

```
PROPERTY <DigitalClockCompID> SHOWSECONDS 'value'
The default is set to 'TRUE'
```

```
-- Example 03: To Disable/Hide the Seconds in Digital Clock
```

```
PROPERTY <DigitalClockCompID> SHOWSECONDS 'FALSE'
```

```
-- Example 04: To Enable/Show the Seconds in Digital Clock
```

```
PROPERTY <DigitalClockCompID> SHOWSECONDS 'TRUE'
```

8.56 Managing RBTI Form Variables

When working on a single form at a time, it is very easy to manage the following RBTI Form Variables:

- RBTI_FORM_FORMNAME (TEXT)
- RBTI_DIRTY_FLAG (INTEGER)
- RBTI_FORM_ALIAS (TEXT)
- RBTI_FORM_MODE (TEXT)
- RBTI_FORM_DIRTYVAR (INTEGER)
- RBTI_FORM_TBLNAME (TEXT)
- RBTI_FORM_COLNAME (TEXT)
- RBTI_FORM_COLVALUE (TEXT)
- RBTI_FORM_DATATYPE (TEXT)
- RBTI_FORM_COMPID (TEXT)

However, if you have a situation where a form is called from within another form, you may have a stack of forms being displayed on top of each other. RBTI Form Variables defined by the last form will of course overwrite the results of the previous form.

So, how do you keep track and manage all resulting variables for each form separately?

To manage this particular situation, all you need to do is define form variables with specific form-related expressions with a unique naming convention.

Here's how:

Assuming you have two forms (FormA and FormB).

1. Define the following expressions for FormA

```
vFormA_FormName TEXT = (.RBTI_FORM_FORMNAME)
vFormA_DirtyFlag INTEGER = (.RBTI_DIRTY_FLAG)
vFormA_Alias TEXT = (.RBTI_FORM_ALIAS)
vFormA_Mode TEXT = (.RBTI_FORM_MODE)
vFormA_DirtyVar INTEGER = (.RBTI_FORM_DIRTYVAR)
vFormA_TblName TEXT = (.RBTI_FORM_TBLNAME)
vFormA_ColName TEXT = (.RBTI_FORM_COLNAME)
vFormA_ColValue TEXT = (.RBTI_FORM_COLVALUE)
```

```
vFormA_DataType TEXT = (.RBTI_FORM_DATATYPE)
vFormA_CompID TEXT = (.RBTI_FORM_COMPID)
```

2. Define the following expressions for FormB

```
vFormB_FormName TEXT = (.RBTI_FORM_FORMNAME)
vFormB_DirtyFlag INTEGER = (.RBTI_DIRTY_FLAG)
vFormB_Alias TEXT = (.RBTI_FORM_ALIAS)
vFormB_Mode TEXT = (.RBTI_FORM_MODE)
vFormB_DirtyVar INTEGER = (.RBTI_FORM_DIRTYVAR)
vFormB_TblName TEXT = (.RBTI_FORM_TBLNAME)
vFormB_ColName TEXT = (.RBTI_FORM_COLNAME)
vFormB_ColValue TEXT = (.RBTI_FORM_COLVALUE)
vFormB_DataType TEXT = (.RBTI_FORM_DATATYPE)
vFormB_CompID TEXT = (.RBTI_FORM_COMPID)
```

Using this technique, all variables will be unique to each individual form, and you can easily track the status of each variable when working in a multi-form application.

8.57 Managing Table Properties on Demand

Did you know that you can manage table properties on demand while using the form in ENTER, EDIT or BROWSE mode?

Imagine a situation where you have one multi-table form, and you wish to manage all related controls for a specific table, based on certain conditions, such as network user.

Conditionally, depending on the logged in user, you may enable or disable all controls related to any table without knowing the individual Component IDs.

Using the "PROPERTY TABLE tablename DISABLECONTROLS" or "PROPERTY TABLE tablename ENABLECONTROLS" commands, you may manage the status of any table as deemed necessary.

Use the following PROPERTY commands as "On After Start EEP":

```
-- To disable all controls related to given table
PROPERTY TABLE tablename DISABLECONTROLS
RETURN

-- To enable all controls related to given table
PROPERTY TABLE tablename ENABLECONTROLS
RETURN
```

Here's a comprehensive list of all supported table properties:

- ENABLE_DELETE_ROWS
- ENABLE_ADD_NEW_ROWS
- DISABLE_DELETE_ROWS
- DISABLE_ADD_NEW_ROWS
- UPDATE
- CLOSE
- DISABLECONTROLS
- ENABLECONTROLS
- OPEN
- EDIT
- NEXT
- PRIOR
- LAST
- FIRST
- INSERT

- APPEND
- DELETE
- POST
- CANCEL
- REFRESH
- JUMPTO
- JUMP
- SEARCH

8.58 Managing Windows Media Player Properties

Did you know that now you can manage every property of the Windows Media Player ActiveX control when implemented in R:BASE forms?

Here is a list of new supported PROPERTY command parameters to manage the Windows Media Player ActiveX control:

Syntax:

```
PROPERTY <WMPActiveXComponentID> 'WMPProperty' 'value'
```

Example 01 (Load File):

```
PROPERTY MediaPlayer 'URL' 'D:\Videos\Stockholm Syndrome.avi'
```

Example 02 (Play):

```
PROPERTY MediaPlayer 'CALL_PROCEDURE Controls.Play' ''
```

Example 03 (Pause):

```
PROPERTY MediaPlayer 'CALL_PROCEDURE Controls.Pause' ''
```

Example 04 (Next):

```
PROPERTY MediaPlayer 'CALL_PROCEDURE Controls.Next' ''
```

Example 05 (Fast Forward):

```
PROPERTY MediaPlayer 'CALL_PROCEDURE Controls.FastForward' ''
```

Example 06 (Fast Previous):

```
PROPERTY MediaPlayer 'CALL_PROCEDURE Controls.Previous' ''
```

Example 07 (Fast Reverse):

```
PROPERTY MediaPlayer 'CALL_PROCEDURE Controls.Reverse' ''
```

Example 08 (Stop):

```
PROPERTY MediaPlayer 'CALL_PROCEDURE Controls.Stop' ''
```

Example 09 (User Interface):

```
PROPERTY MediaPlayer uiMode 'NONE'  
PROPERTY MediaPlayer uiMode 'MINI'  
PROPERTY MediaPlayer uiMode 'FULL'
```

Example 10 (Stretch to Fit TRUE/FALSE):

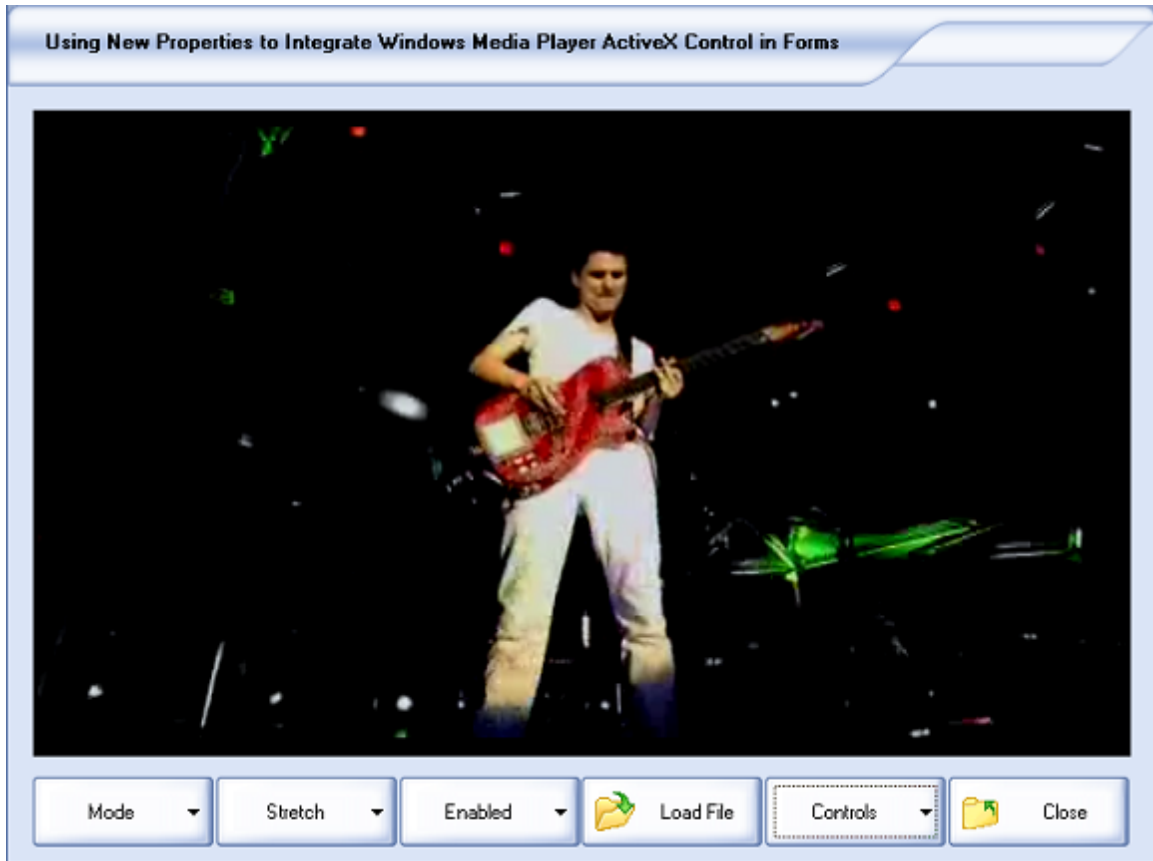
```
PROPERTY MediaPlayer STRETCHTOFIT 'TRUE'
PROPERTY MediaPlayer STRETCHTOFIT 'FALSE'
```

Example 11 (Enabled TRUE/FALSE):

```
PROPERTY MediaPlayer ENABLED 'TRUE'
PROPERTY MediaPlayer ENABLED 'FALSE'
```

Example 12 (Currently Playing File Name)

```
GETPROPERTY MediaPlayer URL vURLName
```



8.59 Moving Form Lookup Variables into Custom EEPs

When upgrading databases from older versions, it is important to take into account if and how R:BASE may work differently. Applications created with legacy logic can perhaps be improved upon in new releases.

For instance, in legacy versions of R:BASE, form variables were not calculated automatically. A `RECALC VARIABLES` command was needed in an EEP to refresh the variables and generate different results. In newer releases of R:BASE (7.x and higher), form variables are recalculated automatically when the cursor moves from field to field.

Because of this logic change in R:BASE, the response time may be longer in forms with many lookup variables which are based upon large tables.

To retain that same performance when using the form, the lookup variables can be populated within an EEP.

In the following variables list, there are 11 lookups performed for the "Client" table based upon a provided client identification number. The variables were meant to display a range read-only information about the client.

Form : OrderEntry

Main Table : Orders

```

1 : TEXT  vClientFirstName = CFirstName IN Client WHERE ClientID = ClientID
2 : TEXT  vClientLastName = CLastName IN Client WHERE ClientID = ClientID
3 : TEXT  vClientCompany = CCompany IN Client WHERE ClientID = ClientID
4 : TEXT  vClientAddress1 = CAddress1 IN Client WHERE ClientID = ClientID
5 : TEXT  vClientAddress2 = CAddress2 IN Client WHERE ClientID = ClientID
6 : TEXT  vClientCity = CCity IN Client WHERE ClientID = ClientID
7 : TEXT  vClientState = CState IN Client WHERE ClientID = ClientID
8 : TEXT  vClientZipCode = CZipCode IN Client WHERE ClientID = ClientID
9 : TEXT  vClientPhone = CPhone IN Client WHERE ClientID = ClientID
10 : TEXT vClientFax = CFax IN Client WHERE ClientID = ClientID
11 : TEXT vClientEmail = CEmail IN Client WHERE ClientID = ClientID

```

Instead, the variable values can be populated using a SELECT command placed within an "On Exit" Custom EEP, within the control where the client ID was entered.

```

-- On Exit EEP
SET VAR vClientFirstName = NULL
SET VAR vClientLastName = NULL
SET VAR vClientCompany = NULL
SET VAR vClientAddress1 = NULL
SET VAR vClientAddress2 = NULL
SET VAR vClientCity = NULL
SET VAR vClientState = NULL
SET VAR vClientZipCode = NULL
SET VAR vClientPhone = NULL
SET VAR vClientFax = NULL
SET VAR vClientEmail = NULL
SELECT +
    CFirstName, +
    CLastName, +
    CCompany, +
    CAddress1, +
    CAddress2, +
    CCity, +
    CState, +
    CZipCode, +
    CPhone, +
    CFax, + +
    CEmail +
INTO +
    vClientFirstName INDIC iv1, +
    vClientLastName INDIC iv1, +
    vClientCompany INDIC iv1, +
    vClientAddress1 INDIC iv1, +
    vClientAddress2 INDIC iv1, +
    vClientCity INDIC iv1, +
    vClientState INDIC iv1, +
    vClientZipCode INDIC iv1, +
    vClientPhone INDIC iv1, +
    vClientFax INDIC iv1, +
    vClientEmail INDIC iv1 +
FROM Client WHERE ClientID = .vClientID

```



```

RECALC VARIABLES
RETURN

```

In the following set of variables lookup data within a "ThirdParty" table based upon a third party payer ID, the variables were also meant to display a range read-only information.

```

12 : TEXT    vTPPContactName = TPPContactName IN ThirdParty WHERE TPP_ID = TPP_ID
13 : TEXT    vTPPCompany = TPPCompany IN ThirdParty WHERE TPP_ID = TPP_ID
14 : TEXT    vTPPContract = TPPContract IN ThirdParty WHERE TPP_ID = TPP_ID
15 : TEXT    vTPPCertificate = TPPCertificate IN ThirdParty WHERE TPP_ID = TPP_ID
16 : TEXT    vTPPNotes = TPPNotes IN ThirdParty WHERE TPP_ID = TPP_ID

```

These variable values can be populated using a similar SELECT command placed within an "On Exit" Custom EEP, within the control where the third party payer ID was entered.

```

SET VAR vTPPContactName = NULL
SET VAR vTPPCompany = NULL
SET VAR vTPPContract = NULL
SET VAR vTPPCertificate = NULL
SET VAR vTPPNotes = NULL
SELECT +
    TPPContactName, +
    TPPCompany, +
    TPPContract, +
    TPPCertificate, +
    TPPNotes ,+
INTO +
    vTPPContactName INDIC iv1, +
    vTPPCompany INDIC iv1, +
    vTPPContract INDIC iv1, +
    vTPPCertificate INDIC iv1, +
    vTPPNotes INDIC iv1 +
FROM ThirdParty WHERE TPP_ID = .vTPP_ID
RECALC VARIABLES
RETURN

```

Other variables which perform calculations must remain in the Form Variables. The variables will continue to be updated if any changes are made to the variables contained within the expressions.

```

17 : TEXT    vDayOfWeek = (SGET((TDWK(DateContarct)),3,1))
18 : CURRENCY vSubTotal = (SUM(ExtPrice))
19 : CURRENCY vFreight = (.vSubTotal * .01)
20 : CURRENCY vSalesTax = (.vSubTotal * .07)
21 : CURRENCY vInvoiceTotal = (.vSubTotal + .vFreight + .vSalesTax)

```

Note:

Using such technique, make sure to predefine all variables with appropriate data type as On Before Start EEP, and clear all necessary variables as On Close EEP.

-- Example

```

-- On Before Start EEP
SET VAR vClientFirstName TEXT = NULL
SET VAR vClientLastName TEXT = NULL
SET VAR vClientCompany TEXT = NULL
SET VAR vClientAddress1 TEXT = NULL
SET VAR vClientAddress2 TEXT = NULL
SET VAR vClientCity TEXT = NULL
SET VAR vClientState TEXT = NULL

```

```
SET VAR vClientZipCode TEXT = NULL
SET VAR vClientPhone TEXT = NULL
SET VAR vClientFax TEXT = NULL
SET VAR vClientEmail TEXT = NULL
SET VAR vTPPContactName TEXT = NULL
SET VAR vTPPCompany TEXT = NULL
SET VAR vTPPContract TEXT = NULL
SET VAR vTPPCertificate TEXT = NULL
SET VAR vTPPNotes TEXT = NULL
RETURN
```

```
-- On Close EEP
CLEAR VARIABLES iv%,vClient%,vTPP%
RETURN
```

You will also need to create two variables as Form Expression to capture the values for entered ClientID and TPP_ID columns.

```
-- Example
nn : INTEGER    vClientID = (ClientID)
nn : INTEGER    vTPP_ID = (TPP_ID)
```

8.60 Persistent Field Value Color in Enhanced DB Grids

When using the color features for Enhanced DB Grids, it is possible to visually compliment the columns based on the column value. However, when using "Row Select" to highlight a specific row, the highlight color for the focused row overwrites the color value for the individual columns.

Did you know a new setting has been introduced to allow the column colors to be persistent, even if a row is selected and highlighted?

A new "Persistent Field Value Color" setting has been implemented which specifies whether the background will not be filled with the Highlight color, when a row is selected. The setting is only recognized when "Row Select" is also checked.

Within the Enhanced DB Grid Properties tab, the "Persistent Field Value Color" setting is located within the control's DB Grid Options.

The screenshot shows the 'Enhanced DB Grid Properties' dialog box with the following settings:

- Size and Coordinates:** Left: 7, Width: 649, Top: 40, Height: 251, Align: None.
- Table:** Table Name: Customer.
- Background Color and Font:** Window, AaBbYyZz.
- DB Grid Options:**
 - Editing
 - Show Editor
 - Titles
 - Indicator
 - Column Resize
 - Column Lines
 - Row Lines
 - Row Select
 - Selection
 - Scroll Hints
 - Multiline Titles
 - Title Ellipsis
 - Column Ellipsis
 - Record Number
 - Cell Hints
 - Allow Tabs
 - Horizontal Scroll Bar
 - Picker for DATE Fields
 - Show Hint
 - Theme Aware
 - Grid Focus Navigation
 - Tab Stop
 - Skip Read Only Columns
 - Multiselect
 - Persistent Field Value Color
- Component ID:** DBGrid_Customers.

Supported Versions:

- R:BASE X (Build...: 10.0.2.11130 or higher)

8.61 Regular Expressions in DB/Variable Edit Controls

Did you know that regular expressions are now supported for Variable Edit and DB Edit controls?

A regular expression is a sequence of characters that define a search pattern, which can be used to validate user input. Usually this pattern is then used by string searching algorithms for "find" or "find and replace" operations on strings.

Within the "Effects" tab of the Variable Edit and DB Edit control properties, a button is also available to validate the defined regular expression.

Examples:

01.

The following can be used to validate an email address:

```
\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,}\b
```

02.

The following can be used to validate any phone number (e.g. 555-6592):

```
\d\d\d\d-\d\d\d\d
```

Additional details regarding regular expressions and supported characters is available within the Forms help file: "Using Regular Expressions"

Supported Versions:

- R:BASE X (Build..: 10.0.2.11227 or higher)

8.62 Searching in the DB Tree View Control

Did you know that you can search for any text in nodes that is dynamically generated by DB Tree View control?

PROPERTY command example:

```
PROPERTY TreeViewControl FINDNODE 'Daily'
```

The DB Tree View control also supports the ability to search the nodes for text, using [Ctrl] +[F] to display a search dialog.

8.63 Selecting All Items in List Box/List View Controls

Did you know that you can use the [Ctrl+A] or [Shift+F6] keys to toggle between selecting all items and unselecting all items in Variable Lookup ListBox or Variable Lookup ListView controls?

Here's how to "Select All Items" in Variable Lookup ListBox and Variable Lookup ListView controls.

1. Start R:BASE.
2. Connect to your database.
3. Open a form with a Variable Lookup ListBox or Variable Lookup ListView Control, in the Form Designer.
4. Make sure that associated variable data type is capable of accepting multi-select List Source Key Field items.
5. Make sure to enable the "MultiSelect" option for the Variable Lookup ListBox or ListView Control.
6. Save the form.
7. Run the form.
8. Use [Ctrl+A] or [Shift+F6] to toggle between Select and Un-Select All items.
9. Use [Ctrl+Click] to select multiple items.
10. Use [Shift+Click] to select an highlighted group of items.

You may also use the following PROPERTY commands to dynamically select all items, if you wish:

```
-- For Variable Lookup ListBox Control
PROPERTY <VarLookupListBoxCompID> MULTISELECT 'TRUE'
PROPERTY <VarLookupListBoxCompID> SELECT_ALL 'TRUE'
RETURN
```

```
-- For Variable Lookup ListView Control
PROPERTY <VarLookupListViewCompID> MULTISELECT 'TRUE'
PROPERTY <VarLookupListViewCompID> SELECT_ALL 'TRUE'
RETURN
```

8.64 Setting the Background Color for Focused Controls

When you design a cool form with so many DB/Variable Edit controls, sometimes the end user simply cannot figure out where the tiny mouse cursor is located. In the past we have found a workaround to change the background color properties for the "On Enter EEP" for each control.

Did you know that you can use the default "Form Settings" for "Edit Objects" to achieve your goal without customizing each and every edit control on the form?

Here's how:

1. Start R:BASE for Windows
2. From the Database Explorer > Main Menu > Settings > Form Designer
3. Within the Default Form Settings > Edit Objects ...

[] Background Color When Focused

This is where you'll have to define the color, including the option to define your own custom color, if you wish.

Using this custom setting will provide the option for any individual to use their favorite color.

You may also customize this setting for individual DB/Variable Edit/Memo controls as follows:

Form Designer > Object Properties > Effects > [] Background Color When Focused

You may use the PROPERTY command as "On After Start EEP" to dynamically enforce/change the background color for any DB/Variable Edit/Memo Controls.

Example:

```
PROPERTY <ComponentID> HighlightFocus 'TRUE'
PROPERTY <ComponentID> HighlightFocusColor 'YELLOW'
RETURN
```

8.65 Storing Comment and Version Information for External Form Files

Did you know that you can store comment and version information for External R:BASE Form Files (.RFF)?

After creating/saving a new external form file, the name, latest revision date, size, and version number will be displayed in the Database Explorer.

A comment can be added and/or updated by right clicking on the external form file and choosing "Change Comment...".

8.66 Support for Native Form Display

A new application property has been implemented allowing forms to be displayed as "native", which eliminates the use of operating system themes on R:BASE forms. With the setting OFF (default), the feature is disabled.

```
PROPERTY APPLICATION NATIVE_FORMS 'ON'
PROPERTY APPLICATION NATIVE_FORMS 'OFF'
```

The following is an R:BASE form example with the NATIVE_FORMS property set to OFF, as the default value.

Customer Management

Customers	Customer Details	
<ul style="list-style-type: none"> Barton and Associates Blue Ridge Technologies, Inc. Bytes & Words Compdat Computer Consulting Compumasters Computer Supply Computer Medical Ctr. Computer Mountain Inc., Computer Warehouse - I <li style="background-color: #e0e0e0;">Data Solutions Datacrafters Infosystems Industrial Computers Inc. Industrial Concepts Inc. Johnson Technologies Manufacturers Discount Computers Microcomputer Distribution Microtech University - I Midtown Computer Co. MIS by Design Modular Software, Inc. Murrysville Technology Center Nordan Distributors, Inc. Olympic Sales Open Systems I/O PC Consultation And Design RAM Data Systems, Inc. Renaissance Computer Company Softech Database Design State University The Data Shop 	<p>Cust ID: <input type="text" value="117"/></p> <p>Company Name: <input type="text" value="Data Solutions"/></p> <p>Address: <input type="text" value="3923 Pleasant Hill Dr"/></p> <p>City, State, Zipcode: <input type="text" value="Olympia, WA 98504"/></p> <p>Phone: <input type="text" value="206-321-5606"/></p> <p>Mod Level: <input type="text" value="2"/></p> <p>Last Update Date: <input type="text" value="10/14/2013"/></p> <p>Last Update Time: <input type="text" value="1:08 PM"/></p> <p>Last Order Date: <input type="text" value="10/15/2010"/></p> <p>Web URL: <input type="text" value="http://www.dsolutions.com"/></p> <p>E-Mail Address: <input type="text" value="sales@dsolutions.com"/></p> <p>Current Status: <input type="text" value="A"/></p> <p>Payment Terms: <input type="text" value="Net 10 Days"/></p> <p>Credit Limit: <input type="text" value="\$10,000.00"/></p>	
<input type="button" value="Add New Customer"/>	<input type="button" value="Edit Selected Customer"/>	<input type="button" value="Finished"/>

The following is an R:BASE form example with the NATIVE_FORMS property set to ON.

Customer Management

Customers	Customer Details
Barton and Associates	Cust ID: <input type="text" value="117"/>
Blue Ridge Technologies, Inc.	Company Name: <input type="text" value="Data Solutions"/>
Bytes & Words	Address: <input type="text" value="3923 Pleasant Hill Dr"/>
Compdat Computer Consulting	City, State, Zipcode: <input type="text" value="Olympia, WA 98504"/>
Compumasters Computer Supply	Phone: <input type="text" value="206-321-5606"/>
Computer Medical Ctr.	Mod Level: <input type="text" value="2"/>
Computer Mountain Inc.,	Last Update Date: <input type="text" value="10/14/2013"/>
Computer Warehouse - I	Last Update Time: <input type="text" value="1:08 PM"/>
Data Solutions	Last Order Date: <input type="text" value="10/15/2010"/>
Datacrafters Infosystems	Web URL: <input type="text" value="http://www.dsolutions.com"/>
Industrial Computers Inc.	E-Mail Address: <input type="text" value="sales@dsolutions.com"/>
Industrial Concepts Inc.	Current Status: <input type="text" value="A"/>
Johnson Technologies	Payment Terms: <input type="text" value="Net 10 Days"/>
Lanufacturers Discount Computers	Credit Limit: <input type="text" value="\$10,000.00"/>
Microcomputer Distribution	
Microtech University - I	
Midtown Computer Co.	
MIS by Design	
Modular Software, Inc.	
Murrysville Technology Center	
Nordan Distributors, Inc.	
Olympic Sales	
Open Systems I/O	
PC Consultation And Design	
RAM Data Systems, Inc.	
Renaissance Computer Company	
Softech Database Design	
State University	
The Data Shop	

Note the different button sizes in the title bar, and the thin form border.

Pros and Cons

The NATIVE_FORMS property is offered as an optimization option to eliminate the theme-based drawing, painting, refreshing, etc., and to improve performance. There is however a functionality trade-off. It is important to note that in native mode some items would be disabled, such as the drop-down menu on the form title bar, or the ability to have only the minimize and X buttons on the title bar. Users may be required to rethink a form's design and presentation when the NATIVE_FORMS property is set to ON.

Native Form Use

The NATIVE_FORMS property is a global value. When the PROPERTY is ON, "any" launched form will be displayed as native. It's up to the user to decide how to manipulate the property. To enable native forms on a per-form basis, the property should be set ON before a form is launched with the REFF, BROWSE, EDIT, and ENTER commands, and then OFF within the form's "On After Start" EEP. If a majority of the forms are to be native, then the property can be set to ON within the application on startup. For example, if there are hundreds of forms that should be native and just a dozen or so to be theme-based, then NATIVE_FORMS can be set to ON within the startup process, and then for each of the forms to be theme-based, the property can be turned OFF before a form is launched and set ON again within the "On After Start" EEP. Thus, setting the property to limit the effect to the forms be launched.

8.67 Suppressing Script Errors in WEB Browser Controls

When implementing Web Browser, DB Web Browser, or Variable Web Browser controls in forms, depending on the server URLs, users may encounter script errors or warning messages which force them to click on an "OK" button to continue and so on. This may create an inconvenience for your R:BASE custom applications.

Now, you can suppress those warning or script error messages by either checking the additional property option to "Suppress Script Errors" for Web Browser, DB Web Browser or Variable Web Browser controls or use the following PROPERTY command:

```
PROPERTY WebBrowserComponentID SuppressScriptErrors 'TRUE'  
RETURN
```

8.68 Tab Order Sequence in the Form Designer

Did you know the tab order of form controls, specifically within the Form Designer and not at runtime, was recently updated to follow the "Tab Order" settings sequence?

Prior to the change, the tab order within the Form Designer follows the alphabetical order of the control names (e.g. RDBEdit2, RDBEdit3, RDBEdit4, etc.). However, since the use of the "name" for the tab order provides no relative use, the Form Designer was updated to use the sequence defined within "Tab Order" settings.

This change allows users to verify the tab order sequence without having to actually launch the form!

Within the Form Designer, select "Layout" > "Change Tab Order", or right click upon form background, or any form container, and select "Change Tab Order" to view the tab order settings.

Supported Versions:

- R:BASE X (Build...: 10.0.2.11130 or higher)

8.69 Understanding Tab Order

When you place any control(s) in a form designer which posses the property of being "Containers" itself, such as Enhanced Panel, Enhanced Group Box, Scrolling Region and Tab Control, then the tab order of any placed controls within is related to that container itself.

Tab Order Priority:

01. Form itself
02. Panel(s) and Controls within that panel
03. Individual Controls

It is also important to know the order in which the Panel(s) were defined.

So, if you would like to re-order the tab within each container (Enhanced Group Box in your case), first you need to select that container and then:

```
Layout > Change Tab Order will display the objects placed on that container, such as Enhanced Group  
Box.
```

Also, the form designer remembers the order in which you placed the container and the contents within that container.

BUT ... using the PROPERTY command, you CAN change the tab order as well as the focus.

Typical Scenario:

Let's say you have three "Enhanced Group Box" controls on the form and when the form starts, you would like the focus to be on the third (3rd) control (DB Edit) on the Middle (2nd) Enhanced GroupBox, irregardless of the sequence they were placed on the form, etc.

In that case:

1. Assign the Component ID as "Panel2" for the middle Enhanced Group Box.
2. Assign the Component ID as "Field3" on the third control (DB Edit) placed on the middle (2nd) Enhanced Group Box.
3. Create an EEP, for example, TabOrder.EEP with the following commands:

```
PROPERTY Panel2 TABORDER 0
PROPERTY Field3 SET_FOCUS 0
RETURN
```

4. Use TabOrder.EEP as "On After Start EEP"

Form Properties > Miscellaneous > "On After Start EEP"

5. Save the form.

Now when you run that form, the focus will be on the third control (DB Edit) placed on the middle (2nd) Enhanced Group Box.

8.70 Understanding the "Space Evenly" Option for Radio Group Controls

Within the Form Designer, choose either the DB RadioGroup or the Variable RadioGroup. View the Properties for the control and select the "Effects" tab.

When "Space Evenly" option is set to True (checked), the columns are positioned so that each "column" of controls takes up the same amount of space. This specific option is very useful when "Columns" are set to more than 1.

Example:

Properties > Columns: 2

To specify the number of pixels used between "rows" (when Columns: is set to 1), use the PROPERTY command to specify the number of pixels used between rows of Radio Buttons in a Radio Group Control.

Example:

```
PROPERTY <Component ID> VERTICALSPACING xxx
```

Where xxx is the number of pixels used between rows of Radio Buttons in a Radio Group Control.

8.71 Using a Date Picker (Calendar) in a DB Grid

Did you know that you can use a Date Picker (Calendar) in a DB Grid?

If you have a DB Grid Column associated with DATE, now you can use the Calendar to Enter or Update the field in DB Grid.

Here's how:

1. Form Designer > DB Grid Control > Properties > Effects
2. [x] Check the option for "Show Picker for DATE Fields" and select the "OK" button
3. Save and close the Form Designer
4. Database Explorer > Select Form > Run Form

Notice the drop-down arrow on DATE field when the date column is focused.

Clicking on the drop-down arrow will bring up a calendar.

8.72 Using a Default State for DB/Variable Check Boxes

Did you know that you can set the "Default State" for DB and Variable Check Boxes?

- Form Designer > DB Check Box > Properties ...

Default State: Unchecked Checked

- Form Designer > Variable Check Box > Properties ...

Default State: Unchecked Checked

8.73 Using Bidirectional Mode Properties in R:BASE

Did you know that you can enable the control to adjust its appearance and behavior automatically when the application runs in a locale that reads from right to left instead of left to right?

A new BiDiMode (Bidirectional Mode) properties have been implemented for the following controls:

- DB Check Box
- DB Radio Button
- DB Radio Group
- Variable Check Box
- Variable Radio Button
- Variable Radio Group

Form Designer | Object Properties | Effects | Bidirectional Mode

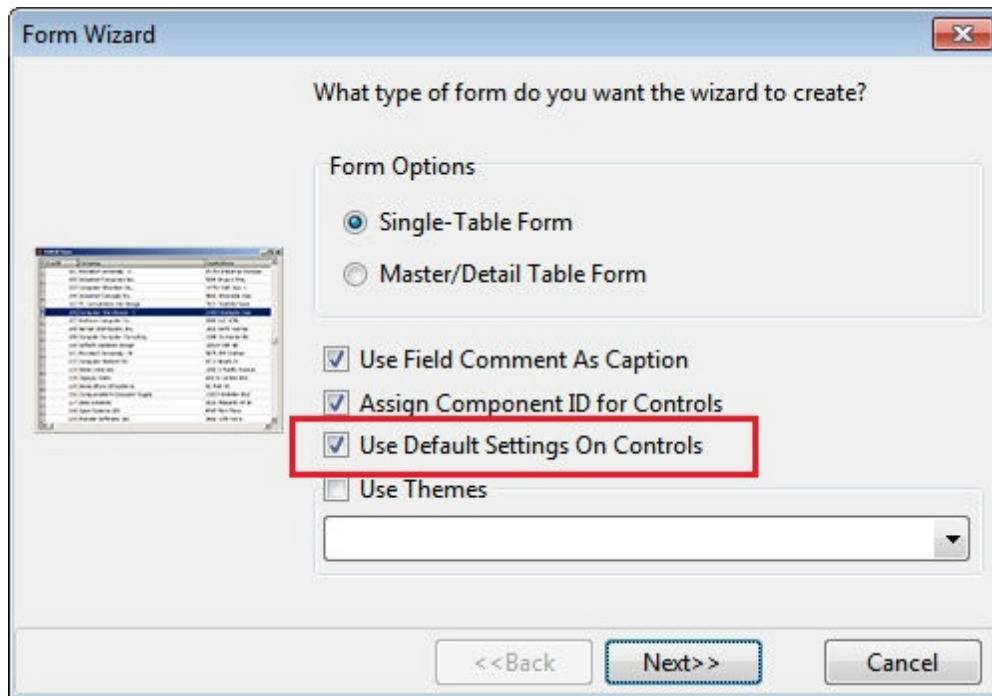
Left to Right Right to Left, No Align

Right to Left Right to Left, Reading Only

The bidirectional mode controls the reading order for the text, the placement of the vertical scroll bar, and whether the alignment is changed.

8.74 Using Form Default Settings for the Form Wizard

Did you know when creating a new form with the Form Wizard, a newly added "Use Default Settings on Controls" check box is available, so the new form will use the customized defaults established by the user/developer.



To customize the default settings for form and controls, select "Settings" > "Form Designer" from the main menu bar.

The defined settings will be enforced for new forms, and newly added controls for the Form Designer, and now the Form Wizard!

Supported Versions:

- R:BASE X (Build...: 10.0.2.11130 or higher)

8.75 Using Mask Options with the File List Box Control

Did you know that many variations of file masks are supported within the File List Box control, to limit the displayed results?

The "File Mask" accepts wildcard values like DOS prompt file masks. The asterisk (*) is a wildcard which matches any number of arbitrary characters. The question mark (?) is a wildcard which matches a single arbitrary character. The file mask *.* displays all files, which is the default value.

For example, when using the question mark, the File List Box will limit the displayed files to JPGs starting with "Image" and has a maximum of eight characters:

Image???.jpg

The asterisk can be used to specify any number of characters before or after the specified text, meaning the following will display images that contain any variation with the text "form" in the file name (e.g. CopyForm.jpg, FormSettings.jpg, etc.):

form.jpg

Multiple masks can also be specified using a semicolon as the mask separator. For example, the following will limit the displayed files to JPG, PNG, and BMP images:

.jpg;.png;*.bmp

To enhance the File List Box to use a mask dynamically, use the PROPERTY command within an EEP as

follows, to display only PDF document that begin with the text "Invoice":

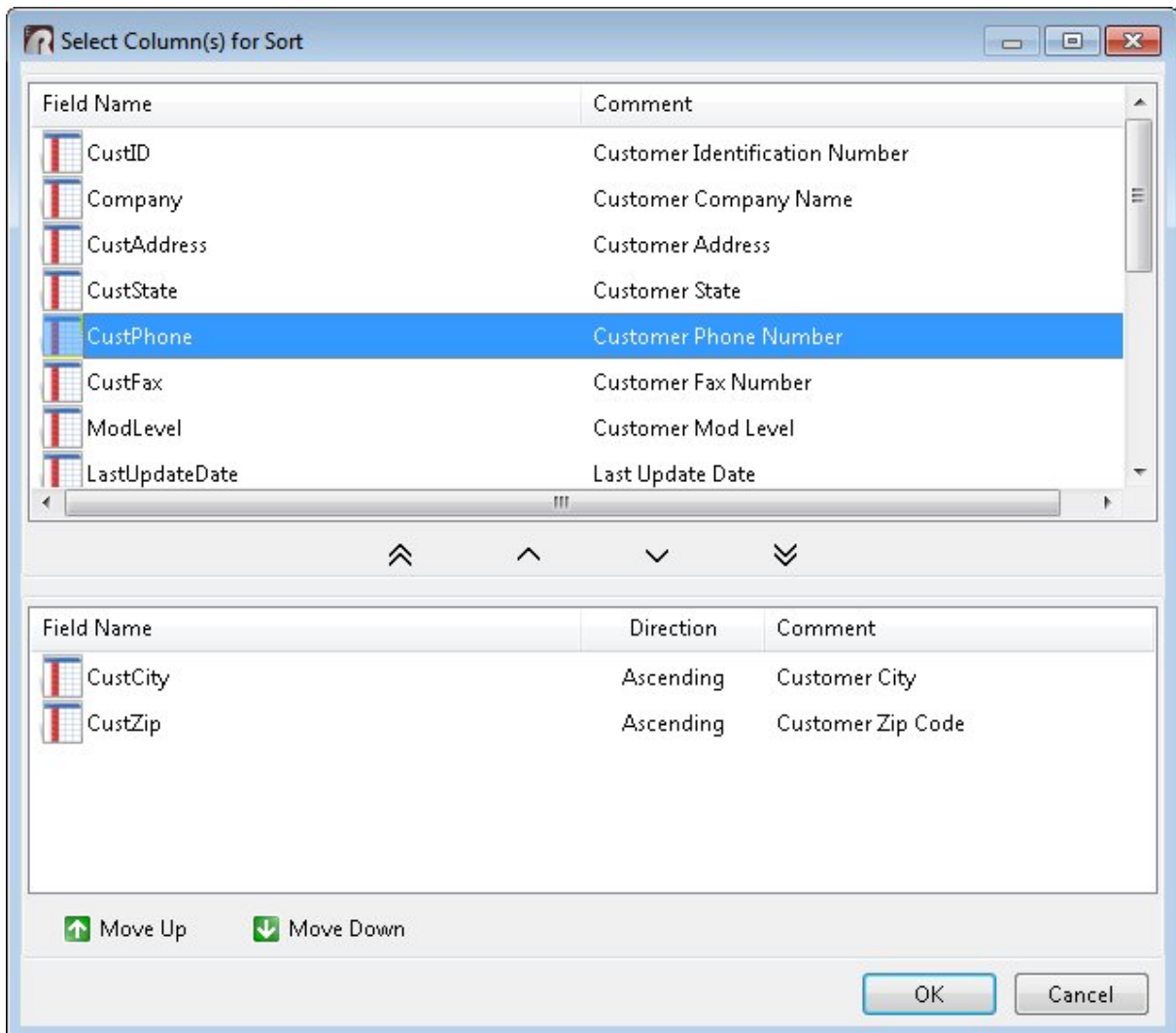
```
PROPERTY FileListBox1 MASK 'Invoice*.pdf'
```

8.76 Using Multi-Column Sort in Enhanced DB Grid Control

Did you know R:BASE includes the option to use Multi-Column Sort in the Enhanced DB Grid Control?

Here's how:

1. Start R:BASE
2. Connect the RRBYW19 database and run the form with Enhanced DB Grid control
3. Use [Ctrl + D] to bring up the screen to select column(s) for sort.



4. Select the column name and then click on [Add Column] button to add to the list.

Repeat the same to add column(s) for additional sort

Tips:

- You may also double-click on the column name to "select and add" to the list of columns to sort.

- Use [Remove Column] button to remove selected column from the list
 - Use [Move Up] or [Move Down] button to re-arrange the list of columns
 - You may also double-click on the column name to "un-select" the column from the list of columns to sort.
5. Once done selecting all columns to sort, click on [OK] button to close the screen.
 6. Notice the Enhanced DB Grid with Sorted Columns
 7. Use [Shift + Ctrl + D] to clear the sort, if you wish.

8.77 Using the "On DB Grid Filter Change" EEP

The Enhanced DB Grid control in the Form Designer allows you to create filters for any column associated with the grid. These filters can be based upon all matching values as you type, or column values shown as unique (grouped) column values. Clicking on the value from the drop-down value will display the updated records in the Enhanced DB Grid.

Did you know that now you can define an "On DB Grid Filter Change EEP" to perform additional routines that may fulfill your additional needs?

Enhanced DB Grid Properties > EEPs > On DB Grid Filter Change EEP

8.78 Using the FILTERTYPE Property in Enhanced DB Grids

While in the Form Designer, the Enhanced DB Grid control includes the option of implementing Column Filters and customizing the Filter Type as CONTAINS, LIKE, or EQUAL.

Enhanced DB Grid Control > Properties > Enhanced Options > Filter Type

Enhanced DB Grid Control > Properties > Columns > Column Properties > Filter

Did you know that now you can dynamically change the filter type using the newly introduced FILTERTYPE property?

The following PROPERTY command parameter is now supported:

Syntax:

```
PROPERTY <EnhancedDBGridComponentID> FILTERTYPE 'value'
```

Examples:

```
PROPERTY <EnhancedDBGridComponentID> FILTERTYPE 'EQUAL'  
PROPERTY <EnhancedDBGridComponentID> FILTERTYPE 'LIKE'  
PROPERTY <EnhancedDBGridComponentID> FILTERTYPE 'CONTAINS'
```

Have fun implementing Filters and Filter types using the Enhanced DB Grid control!

8.79 Using the Enhanced DB Grid Sort Column Properties

Did you know that you can sort the Enhanced DB Grid Columns without clicking on the actual column header, using the PROPERTY commands?

Here's how:

```
PROPERTY <EnhancedDBGridComponentID> SortAsc '0'  
PROPERTY <EnhancedDBGridComponentID> SortDesc '0'
```

-- Examples (Based on RBG9_EnhancedDBGrid form included with RRBYW19)

```
-- To Dynamically Sort (ASC) the first column (Company)
PROPERTY DBGrid_Customers SortAsc '0'
```

```
-- To Dynamically Sort (DESC) the first column (Company)
PROPERTY DBGrid_Customers SortDesc '0'
```

```
-- To Dynamically Sort (ASC) the second column (City)
PROPERTY DBGrid_Customers SortAsc '1'
```

```
-- To Dynamically Sort (DESC) the second column (City)
PROPERTY DBGrid_Customers SortDesc '1'
```

Alternative, you could also use the following PROPERTY syntax as well.

```
PROPERTY <EnhancedDBGridComponentID> SortAsc 'columnname'
PROPERTY <EnhancedDBGridComponentID> SortDesc 'columnname'
```

```
-- To Dynamically Sort (ASC) the first column (Company)
PROPERTY DBGrid_Customers SortAsc 'Company'
```

```
-- To Dynamically Sort (DESC) the first column (Company)
PROPERTY DBGrid_Customers SortDesc 'Company'
```

```
-- To Dynamically Sort (ASC) the second column (City)
PROPERTY DBGrid_Customers SortAsc 'City'
```

```
-- To Dynamically Sort (DESC) the second column (City)
PROPERTY DBGrid_Customers SortDesc 'City'
```

8.80 Using the Mouse Over Property for Images

Did you know that in addition to "On Click" and "Right Click" EEPs, now you also have the following additional properties for Image, DB Image and Variables Image Controls?

- On Mouse Enter
- On Mouse Over
- On Mouse Leave

On Mouse Enter:

Use this option to run EEP automatically when the mouse cursor moves into the control's area.

On Mouse Over:

Use this option to run EEP when the mouse moves across the control. Please keep in mind that as the mouse cursor moves across a control, the EEP is called repeatedly. Each time it is called, it is with the new coordinates that reflect the continuous path of the mouse cursor across the screen real estate covered by the control's visual representation.

On Mouse Leave:

Use this option to run EEP automatically when the mouse cursor moves out of the control's area.

8.81 Using the New Display Format in Forms

Generally, all DB and Variable controls in Form Designer include a "Format Mask" property option to customize the format for data entry and data update purposes. This Format Mask property cannot be used for customizing the "display format" of database or variable edit controls.

Did you know that you can use the newly implemented enhancement to customize the "Display Format" for DB and Variable controls in form designer?

Here's how:

1. Open any form in the Form Designer
2. From the Menu Bar, choose "Tables" > "Display Format ..."

Notice the dynamic Tables/Columns Tree View with all associated table(s) and the list of customizable column(s).

3. Click on the desired column that needs the custom display format.
4. Depending on the data type of the selected column, a list of appropriate display formats is generated.
5. Select the most suitable format from the list. You may further customize (overwrite) the selected format is necessary.
6. Click on [OK] button to save the Display Format.

Using this enhanced feature, now you can have your cake and eat it too!

8.82 Using the On Increment/On Decrement EEPs for Spinner Controls

Did you know that you fire an Entry/Exit Procedure (EEP) for a DB Spinner and Variable Spinner control each time the value is increased or decreased?

The Spinner controls now include the "On Increment" and "On Decrement" EEP options that will run the specified EEP once the action is taken.

The available EEPs now include:

On Double Click EEP

Double clicking on the control will run the specified EEP

On Increment EEP

Increasing the value for the control will run the specified EEP

On Decrement EEP

Decreasing the value for the control will run the specified EEP

8.83 Using the "On Right-Click" EEP on List Box/List View Controls

Did you know that you can use the "Right-Click" mouse button on Variable Lookup List Box or Variable Lookup List View controls to execute any code, defined within a "On Right-Click" EEP?

Here's how:

For the Variable Lookup List Box:

1. Within the Form Designer, select an existing or add a new Variable Lookup List Box
2. Within the Object Properties, select the "EEPs" tab.
3. Notice the additional option for "On Right Click"
4. You may also access your Custom EEP by simply right-clicking on the object and selecting the "Custom EEP" > "On Right Click" menu option.

For the Variable Lookup List View:

1. Within the Form Designer, select an existing or add a new Variable Lookup List View
2. Within the Object Properties, select the "EEPs" tab.
3. Notice the additional option for "On Right Click"
4. You may also access your Custom EEP by simply right-clicking on the object and selecting the "Custom EEP" > "On Right Click" menu option.

8.84 Using the SCALEBY Property

Did you know that you can scale the form and all placed control dynamically to fit the screens with higher resolution?

You may use the `PROPERTY RBASE_FORM SCALEBY 'nnnn'` command to automatically scale form and all related controls to specified screen pixel width.

SCALEBY property modifies the Height and Width properties of the form and all controls placed on it, but it does not change the Top and Left properties; therefore, the size of the controls changes, but not the coordinates of the top-left corners of the controls. Thus, it preserves the position of all controls on the form.

If you have developed a form that is 800x600 for the majority of users having a monitor with 800x600 screen resolution to fit the form on the entire available screen, but a few users of your application have a higher resolution (1024x768) and if you wish your 800x600 form to fill the entire screen of a monitor with a resolution of 1024x768, you'll need to use the `RBASE_FORM SCALEBY` property as follows:

```
PROPERTY RBASE_FORM SCALEBY '800 '  
RETURN
```

This will scale the original form designed as 800x600 to fill the entire screen even if the screen resolution is higher than 800x600, such as, 1400x1050, 1280x800 or 1024x768.

Notes:

- Use the SCALEBY property as embedded custom EEP in "On After Start EEP" in form
- To take advantage of this feature, make sure that you use the scaled true type fonts for all controls on that form

8.85 Using the File List Box Control

Did you know that now you can use "On Click EEP" for the File List Box Control?

Form Designer > File System Controls > File List Box > Properties

Using this feature, you can capture the entire file name as a "Long File Name" or "Short File Name" into variable(s) by simply clicking on any file name listed in the File List Box.

Example:

1. Place a File List Box Control on a form.
2. Assign a Component ID for File List Box Control, for example, "cFileListBoxControl".
3. Place two Variable Edit or Variable Label Controls associated with "vLongFileName" and "vShortFileName" to monitor the captured results.
4. Use the following code as File List Box Control "On Click EEP":

```
-- Begin  
-- File List Box Control (On Click Custom EEP)  
CLEAR VAR vLongFileName, vShortFileName  
SET VAR vLongFileName TEXT = NULL  
SET VAR vShortFileName TEXT = NULL  
GETPROPERTY cFileListBoxControl LONGFILENAME vLongFileName  
GETPROPERTY cFileListBoxControl SHORTFILENAME vShortFileName  
RECALC VARIABLES  
RETURN  
-- End
```

5. Save and close the form.
6. At the R> Prompt or using the Database Explorer:


```
EDIT USING formname
```

Click on any file listed in the list box and notice the resulting captured values as Long/Short file names.

8.86 Using the Directory Tree Control

Did you know that now you can use an "On Click EEP" for the Directory Tree Control?

Form Designer > File System Controls > Directory Tree > Properties

Using this feature, now you can capture the value into a variable by simply clicking on any node of the tree.

Example:

1. Place a Directory Tree Control on a form.
2. Assign a Component ID for Directory Tree Control, for example, "cDirTreeControl".
3. Place a Variable Edit or Variable Label Control on the form to monitor the captured results.
4. Use the following code within the Directory Tree Control "On Click EEP":

```
-- Begin
-- Directory Tree Control (On Click Custom EEP)
CLEAR VAR vValue
SET VAR vValue TEXT = NULL
GETPROPERTY cDirTreeControl DIRECTORY vValue
RECALC VARIABLES
RETURN
-- End
```

5. Save and close the form.
6. At the R> Prompt or using the Database Explorer:

```
EDIT USING formname
```

Click on any level of the Directory Tree Control and notice the resulting captured value.

8.87 Using the Multi-Select Option for the List View Control

Did you know that you can use a multi-select option for the Variable Lookup List View control in forms?

Here's how:

Form Designer > Variable Controls > Variable Look-up List View > Object Property > Effects > MultiSelect

Notes:

1. To use the multi-select option, make sure that the variable data type is pre-defined as TEXT, NOTE or VARCHAR. Choosing which of these to use depends on the anticipated cumulative value for the variable. You must choose the data type wisely as the TEXT variable length is limited to 1500 characters, the NOTE variable length is limited to 4092 characters, and the VARCHAR variable length is limited to 256MB.
2. Use single-click to select one item.
 - Use [Ctrl + Click] to add item(s) to the list
 - Use [Ctrl + Click] to un-select the already selected item
 - Use [Ctrl + A] to select ALL items
 - Use [Ctrl + A] to un-select ALL items
 - Use [Shift + F6] key to select ALL items
 - Use [Shift + F6] key to un-select ALL items

3. All key values will be separated by a comma.

8.88 Using the Object List Toolbar

While in Form Designer, the Object List is very useful to displays the list of *all* object(s) placed on the form. You may view all objects by selecting [All Controls] from the Object List, or view all objects by selecting individual category, such as:

- Standard Controls
- Database Controls
- Variable Controls
- Additional Controls
- Internet Controls
- File System Controls
- Legacy Controls
- ActiveX Controls

By selecting the object and clicking on [Object Property] button, you may customize the property of selected object.

Did you know that you can also dock or float the Object List? Here's how:

Form Designer > Layout > Toolbars > Object List

Notice the Object List.

You may resize the height and width of this control and then dock it or float it, as you wish!

This is similar to other Toolbars.

8.89 Using the "On Key Press" EEP for Instant Results

Did you know that you can use the new "On Key Press" EEP to execute code while you start typing a character in a DB Edit or Variable Edit control?

Use "On Key Press" EEP, an event handler to make something happen as a result of a single character key press on the keyboard.

The key parameter in the "On Key Press" event handler is of type TEXT; therefore, the "On Key Press" event registers the ASCII character of the key pressed. Keys that don't correspond to an ASCII character value (i.e. [Shift] or [F1]) will not execute an "On Key Press" EEP. Key combinations of non-ASCII and ASCII characters (i.e. [Shift]+[A]), execute an "On Key Press" EEP only once. For this example, [Shift]+[A] results in a key value of "A", if [Caps Lock] is off.

8.90 Using the Show Accelerator Character Setting

Did you know that you can use the ampersand character "&" for captions to be used as the ampersand itself character, and not as the "_" accelerator character?

Technically, the ampersand in captions is used for the accelerator character when using the [Alt]+Letter as a hot key (accelerator).

A new "Show Accelerator Character" setting has been added to show "&" as accelerator character "_" or "&" as is, if you wish.

The following form controls properties are enhanced with such setting:

- Push Button
- Bit Button
- Speed Button

- Office Button
- Drop-Down Menu Button
- Enhanced Speed Button
- Enhanced Tab Control

However, if you wish to display the "&" character as is, you will have to un-check the default (checked) "Show Accelerator Character" setting.

Form Designer > Push Button > Object Properties > Attributes ... [] Show Accelerator Character

Form Designer > Bit Button > Object Properties > Attributes ... [] Show Accelerator Character

Form Designer > Speed Button > Object Properties > Attributes ... [] Show Accelerator Character

Form Designer > Office Button > Object Properties > Attributes ... [] Show Accelerator Character

Form Designer > Drop-Down Menu Button > Object Properties > Attributes ... [] Show Accelerator Character

Form Designer > Enhanced Speed Button > Object Properties > Attributes ... [] Show Accelerator Character

Form Designer > Enhanced Tab Control > Object Properties > Attributes ... [] Show Accelerator Character

8.91 Using Zebra Stripes in List View Controls

Did you know that now you can totally customize the Variable Lookup List View control in a form to display Zebra Stripes and custom column titles?

R:BASE includes properties to support "Zebra Stripes" as well as the options to select Background Color and Font Color for Column Titles.

For Zebra Stripes in List View:

1. Open the R:BASE Form Designer
2. Select the List View Control
3. Right click and select Object Property
4. Select the "Effects" tab
5. Check the option for Zebra Stripe and then select the Odd and Even colors.

For Custom Column Headers in List View:

1. Open the R:BASE Form Designer
2. Select the List View Control
3. Right click and select Object Property
4. Select the "Columns" tab
5. Select the column and then define the Background Color and Font Color.

Using these added properties and options, you can customize the List View controls to match your application theme.

Part

IX

9 Reports and Labels

9.1 Altering the CrossTab Grid Color

Did you know that you can customize the grid color of CrossTab control in Reports?

In addition to customizing the CrossTab column and row titles, now you may also customize the CrossTab grid color.

Here's how:

1. Start R:BASE
2. CONNECT to RRBYW19
3. From the Database Explorer > Reports > CrossTab > Design
4. From the Report Designer, right click on the CrossTab control and select "Grid Color..."
5. Select your favorite color and click on "OK" button
6. Click on the "Report Preview" tab to preview the report while in the Report Designer
7. Save the report and close Report Designer

You may also use the PROPERTY command to dynamically change the CrossTab grid color on demand.

Syntax:

```
PROPERTY CrossTabCompID GRIDCOLOR ColorName
```

-- Example 01:

```
PROPERTY CrossTab GRIDCOLOR 'MAROON'  
RETURN
```

-- Example 02:

```
PROPERTY CrossTab GRIDCOLOR 'GREEN'  
RETURN
```

Make sure to use above PROPERTY command as a "Before Generate..." action in reports.

9.2 Capturing Report System Variables

The following System Variables are supported in Reports:

- Date (Current Date)
- Date Time (Current Date and Time)
- Document Name (Report Name)
- Print Data Time (Print Date and Time)
- Page Count (Total page count such as 9)
- Page Set (such as 1 of 9, 2 of 9)
- Page Set with Description (such as Page 1 of 9, Page 2 of 9)
- Page Number (such as 1, 2, 3, 4, 5, 6, 7, 8, 9)
- Page Number with Description (such as Page 1, Page 2, Page 3)
- Time (Current Time)

Did you know that you can capture the value any of these System Variables using the GETPROPERTY command?

Syntax:

```
GETPROPERTY <ReportSystemVariableComponentID> TEXTVALUE varName
```

All you need to do is assign a unique Component ID to a System Variable object placed on a report band and then use GETPROPERTY command to capture the value. In addition, you may also use the following GETPROPERTY commands to retrieve REPORT specific values.

```
-- To Retrieve Report Name
GETPROPERTY REPORT REPORT_NAME vReportName
Report name may also be captured using the RBTI_REPORT_NAME variable.

-- To Retrieve Total Number of Tables Associated with Report
GETPROPERTY REPORT REPORT_TABLE_COUNT vReportTblCount

-- To Retrieve the Name of Master Table
GETPROPERTY REPORT 'REPORT_TABLE_NAME[0]' vReportMasterTbl
```

9.3 Capturing the Report Page Number

Did you know that you can capture the current Report Page Number of the report regardless of any objects on the report Detail Band?

Using R:BASE, you can use the following GETPROPERTY command to achieve such goal.

Here's how:

```
GETPROPERTY REPORT PAGENO vPageNo
RETURN
```

New GETROPERTY REPORT PAGENO vPageNo will return the current page number into a variable (vPageNo) which you can use to accomplish your objective of capturing the report page number.

9.4 Changing the Report Section for All Expressions At Once

Traditionally, while in Report Designer Expression Builder, if you need to change the [Section] of all defined variables, or a group variables, you will have to first highlight the variables, one at a time, and then select the appropriate section from the drop-down list of sections of [Calculate On], and click on [Apply] to change the section.

Did you know that you can select/highlight multiple variables, and then select the appropriate section from the drop-down list of sections of [Calculate On], and click on [Apply] to change section for all expressions at once?

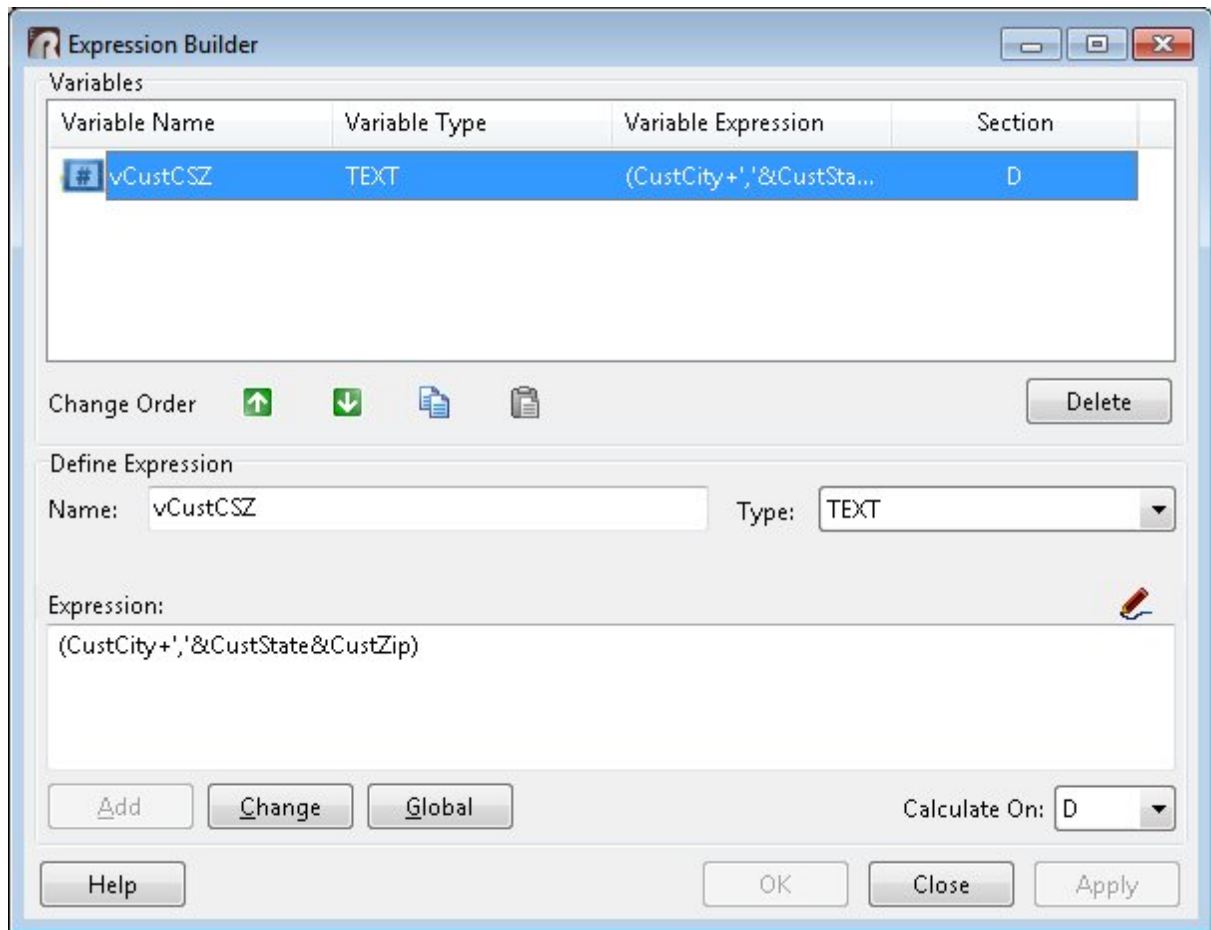
9.5 Copy and Paste Expression Builder Variables

Did you know that you can copy a report/label variable from the Expression Builder, and paste it into another report/label?

Buttons for "Copy" and "Paste" have been added to the Expression Builders for the Form, Report, and Label modules, to allow users to copy and paste selected, or all, variables.

Please note the "Copy" and "Paste" buttons are located next to the "Up" and "Down" arrow buttons, which reorder the location of a variable in the Expression Builder.

In addition, you may also use [Ctrl+C] to copy selected expression(s) and [Ctrl+V] to paste expression(s) while placing the cursor over the expression builder window.



Variables and Expressions can now be easily moved from form to form, label to label, and report to report!

9.6 Customizing Column and Row Titles for CrossTab Reports

Did you know that you can customize the "Titles" for Rows and Columns used in Cross Tab Reports?

Here's how:

1. Report Designer > Cross Tab Control > Configure ...
2. Cross Tab Configuration Layout Panel
3. Right-click on the default column heading and select "Title..." from the list of speed menu options
4. Enter the text for title and click on the [OK] button
5. Right-click on the default row heading and select "Title..." from the list of speed menu options
6. Enter the text for title and click on the [OK] button
7. Use the Steps 3-6 to also customize the following:
 - Font...
 - Color...
 - Display Format...
 - Skip NULLs
 - Gridlines

9.7 Customizing Print Preview Screens with Themes

Did you know that you can customize the Print Preview Screen to match your application/form theme?

Try the following examples and see what you find.

Example 01:

```
CONNECT RRBYW19
CLS
PRINT Invoice WHERE TransID = 1002 +
  OPTION SCREEN +
  |WINDOW_STATE MAXIMIZED +
  |ZOOM_TYPE PERCENTAGE +
  |ZOOMPERCENT 90 +
  |THEMENAME Razzmatazz +
  |PREVIEW_BACKGROUND_COLOR 16180442 +
  |PREVIEW_CAPTION Running R:BASE Your Way!
RETURN
```

Example 02:

```
CONNECT RRBYW19
CLS
PRINT Invoice WHERE TransID = 1002 +
  OPTION SCREEN +
  |WINDOW_STATE MAXIMIZED +
  |ZOOM_TYPE PERCENTAGE +
  |ZOOMPERCENT 90 +
  |THEMENAME Razzmatazz +
  |PREVIEW_BACKGROUND_COLOR [R234,G234,B234] +
  |PREVIEW_CAPTION Running R:BASE Your Way!
RETURN
```

9.8 Defining Built-in PAUSE Messages in Reports

There are situations where a report with hundreds of thousands of records may take a while to print to a file or the screen, and you may want your end-users to stay tuned while the report is being prepared.

The traditional approach of utilizing the PAUSE 3 command in an application before Printing the report is good. However, as soon as the report starts printing, the report takes over control of the window, and your good old PAUSE command is no longer staying on top. So, how do we print a huge report and display the PAUSE message until the entire report is actually generated?

Did you know that you can use a cool PAUSE 3 message within a Report Action, such as the "Before Generate..." action and then issue the "CLS" command to clear the PAUSE 3 message when the report is actually finished in an "After Generate..." action?

Here's how:

1. Start R:BASE for Windows and open your favorite report in the Report Designer
2. Choose "Report" > "Actions" > "Before Generate..." from the main Menu Bar
3. Use the following code in your "Before Generate..." action:

```
-- Before Generate Action in Report
CLS
```



```
GETPROPERTY REPORT REPORT_NAME vReportName
PAUSE 3 USING +
'Preparing Report ... Please Stand By ...' +
CAPTION .vReportName +
ICON APP +
OPTION GAUGE_VISIBLE ON +
| GAUGE_COLOR 2708091 +
| GAUGE_INTERVAL 10 +
| MESSAGE_FONT_NAME VERDANA +
| MESSAGE_FONT_SIZE 10 +
| MESSAGE_FONT_COLOR RED
RETURN
```

4. Choose "Report" > "Actions" > "After Generate..." from the main Menu Bar
5. Use the following code in the "After Generate..." action

```
-- After Generate Action in Report
CLS
CLEAR VARIABLE vReportName
RETURN
```

6. Save the report

You may "Preview" the report while in the Report Designer, PRINT the report from within the Database Explorer or use the PRINT command to see the results.

Notice the real-time PAUSE 3 message with the report name as the caption and the moving gauge, all while the message stays on screen until the entire report is completely generated.

9.9 Documenting Report/Label EEPs

Did you know that while in Report/Label Designer, you can document all Embedded Custom EPPs (if defined), as well as External EEPs (if referenced)?

In addition to documenting Custom EEPs (if defined), now you may also document all External EEPs (if referenced). This will help documenting all references related to Form External EEPs when converting legacy databases and applications.

Here's how:

With the Report Designer, select "Report" > "Document Custom EEPs" from the main Menu Bar, and the available output options include:

- Printer
- Text File
- PDF File
- Clipboard
- **In Text Editor**
- **EEP Map**

With the Label Designer, select "Label" > "Document Custom EEPs" from the main Menu Bar, and the available output options include:

- Printer
- Text File
- PDF File
- Clipboard
- **In Text Editor**
- **EEP Map**

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

9.10 Enabling Consolidated Control Palette in Report/Label Designer

Did you know that you can consolidate all individual Report/Label Designer Controls, namely Standard Controls, Database Controls, Advanced Controls, Variable Controls, and PDF Form Controls as one consolidated Control Palette?

R:BASE includes an option to use the Control Palette for Report/Label Designer. While in Report/Label designer, if you wish to use consolidated Control Palette, follow the steps below:

View | Toolbars | Control Palette

You may grab and dock the Control Palette to the top of bottom of the designer window.

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

9.11 Getting the Total Number of Pages in a Report

Did you know that using the GETPROPERTY command, now you can get the total number of pages in a report?

Here's how:

Use the following custom EEP as an "After Generate EEP..." for the Report Footer:

```
SET VAR vTotPageCount TEXT = NULL
GETPROPERTY REPORT TotalPageCount 'vTotPageCount'
SET VAR vPauseMessage = ('Total Number of Pages:&.vTotPageCount)
PAUSE 2 USING .vPauseMessage +
  CAPTION 'GETPROPERTY in Reports' +
  ICON INFO +
  BUTTON 'Press any key to continue ...' +
  OPTION BACK_COLOR WHITE +
  |MESSAGE_COLOR WHITE +
  |MESSAGE_FONT_COLOR GREEN
CLS
RETURN
```

9.12 Managing the Memo Control

While in Report Designer, after placing a Memo Object on appropriate band, now you can use the double-click to open Memo Editor. You may use the [Load] option to load an external file, if you wish. Clicking on the [OK] button will save the contents. Then, you will be able to preview Memo contents without using the Memo Editor.

You may still use the right-click speed menu for more options related to Memo Control in Report Designer.

9.13 Printing a Portion of a Report to a Different Printer

Did you know that the Sub-Reports have the capability to change the printer settings for the portion of the report in which they generate?

Here's how:

Use the Sub-Report's Printer Setup property to specify the different settings, including a different printer or paper tray, etc., just as you would a normal report.

If you specify a different printer in Printer Setup than that of the main (parent) report, then a New Print Job will be started on that printer when the report generates. This means that different sections of a report can not only use different printer settings, but that they can also be sent to different printers.

To do this, click on the appropriate Sub-Report tab in report designer and then simply customize the page setup as you would do for the normal report.

Report Designer > Select Sub-Report > Main Menu ...

Main Menu > File > Page Setup ... > Printer ...

When the Page Setup property is set for the Sub-Report, then this property determines whether a new print job is started when the Sub-Report is sent to the printer. This property can be used when a portion of a report needs to be printed to a separate printer. If you need to change printer settings for a certain section, you can do so without starting a new print job. Simply set the Page Setup properties (Printer, Paper Size, Orientation, Paper Tray, etc.) and these settings will be applied as the report prints.

Think about printing a Summary Page with R:Charts Analysis on a color printer and then printing all details with hundreds of pages on a separate black and white printer.

9.14 Printing Watermark Images in RTF Reports

When printing an rich text report with transparent images as watermark, everything looks good on the SCREEN. However, if you print that exact report directly to PRINTER, you will notice that the transparency is lost. This is how the rich text reports are printed when sent directly to the printer. This is a limitation of the rich text file format when using transparent images and overlapping text.

In order to print an rich text report with transparent images as watermark, first you need to print that report as PDF and then print that PDF report to printer to achieve your goal. Here's how:

```
PRINT reportname OPTION PDF +
|FILENAME filename.PDF +
|RICHTEXT_ENCODING_TYPE PLAINTEXT +
|OPEN ON
```

9.15 Retrieving Report Properties

Using the GETPROPERTY command you can retrieve the following properties of a report.

- Report Name
- Associated Number of Tables
- Associated Names of Tables (Master/Slave)

(For Sub-Reports)

Example 01: (To Retrieve Report Name)

```
GETPROPERTY REPORT 'REPORT_NAME' vReportName
RETURN
```

Variable vReportName will return the name of the report.

Example 02: (To Retrieve the Number of Associated Tables in Report)

```
GETPROPERTY REPORT 'REPORT_TABLE_COUNT' vReportTableCount
RETURN
```

Variable vReportTableCount will return the total number of associated tables in report.

Example 03: (To Retrieve the Name of Master Table Name for the Report)

```
GETPROPERTY REPORT 'REPORT_TABLE_NAME[0]' vReportMasterTable
RETURN
```

Variable vReportMasterTable will return the name of master table name for the report.

Example 04: (To Retrieve the Name of First Slave Table Name for the Report)

```
GETPROPERTY REPORT 'REPORT_TABLE_NAME[1]' vReportSlaveTable1
RETURN
```

Variable vReportSlaveTable1 will return the name of first slave table associated with a report in a Sub-Report environment.

Example 05: (To Retrieve the Name of Second Slave Table Name for the Report)

```
GETPROPERTY REPORT 'REPORT_TABLE_NAME[2]' vReportSlaveTable2
RETURN
```

Variable vReportSlaveTable2 will return the name of second slave table associated with a report in a Sub-Report environment.

9.16 Sorting Slave Tables with Forms/Sub-Reports

If you have a form with slave tables and a report with sub-reports, and if you wish to sort the matching rows related to slave table(s)/sub-report(s), you'll need to use the ARRANGE BY clause to achieve your goal.

ARRANGE tablename BY columnlist sorts the rows displayed by a form's or report's lower-level tables. Each lower-level table in the form or report that you want to sort requires its own ARRANGE clause. You can specify up to twenty-five tables to ARRANGE and up to five columns in each table on which R:BASE will sort.

Example 01:

```
Database: RRBYW19
Sample Form: SalesTransactions
Associated Tables (5):
  • Customer (Master)
  • InvoiceHeader (Slave 1)
  • InvoiceDetail (Slave 2)
  • Employee (Slave 3)
  • Contact (Slave 4)
```

Command:

```
EDIT USING SalesTransactions +
  ARRANGE InvoiceHeader BY TransID, +
  ARRANGE InvoiceDetail BY ItemNum, +
  ARRANGE Employee BY EmplName,EmpFName +
  ARRANGE Contact BY ContLName,ContFName +
```

```
ORDER BY Company WHERE CustState = 'CA'  
RETURN
```

Example 02:

Database: RRBYW19
Sample Report with Sub-Report: CustomerContacts
Associated Tables (2):

- Customer (Master)
- Contact (Slave 1)

Command:

```
PRINT CustomerContacts +  
ARRANGE Contact BY ContLName,ContFName +  
ORDER BY Company WHERE CustState = 'CA' +  
OPTION SCREEN|WINDOW_STATE MAXIMIZED +  
|ZOOM_TYPE PERCENTAGE|ZOOMPERCENT 90  
RETURN
```

9.17 Using BRUSH_STYLE and PEN_STYLE Properties in Reports

While in Report Designer, when placing a Shape Object, you may also customize the Shape Type, Fill Color, Line Color, Line Thickness, and Line Style.

Did you know that using the new PROPERTY commands you may dynamically change the Shape Style and Pen Style as well?

Syntax:

```
PROPERTY <ShapeObjectCompID> BRUSH_STYLE 'value'
```

Supported Values:

```
SOLID  
CLEAR  
HORIZONTAL  
VERTICAL  
F_DIAGONAL  
B_DIAGONAL  
CROSS  
DIAG_CROSS
```

```
PROPERTY <ShapeObjectCompID> PEN_STYLE 'value'
```

Supported Values:

```
SOLID  
DASH  
DASH_DOT  
DASH_DOT_DOT  
CLEAR  
INSIDE_FRAME  
USER_STYLE  
ALTERNATE
```

You may dynamically change BRUSH_STYLE and PEN_STYLE Properties as On Before Generate Report Action or conditionally as On Before Generate Action on any Report Band, as you wish.

```

-- Example 01
-- Database: RRBYW19
-- Report: InvoiceSummary_Brush_Style_Pen_Style
-- Using New BRUSH_STYLE and PEN_STYLE Properties for Shape Control
-- Action: Report Header [Before Generate]
-- Cover Page Shapes
PROPERTY CoverPage_Shape_01 BRUSH_STYLE 'HORIZONTAL'
PROPERTY CoverPage_Shape_01 PEN_STYLE 'SOLID'
PROPERTY CoverPage_Shape_02 BRUSH_STYLE 'VERTICAL'
PROPERTY CoverPage_Shape_02 PEN_STYLE 'DASH'
PROPERTY CoverPage_Shape_03 BRUSH_STYLE 'F_DIAGONAL'
PROPERTY CoverPage_Shape_03 PEN_STYLE 'DASH_DOT'
PROPERTY CoverPage_Shape_04 BRUSH_STYLE 'B_DIAGONAL'
PROPERTY CoverPage_Shape_04 PEN_STYLE 'DASH_DOT_DOT'
RETURN
-- Action: Page Header [Before Generate]
-- Page Header Shapes
PROPERTY PageHeader_Shape_01 BRUSH_STYLE 'CROSS'
PROPERTY PageHeader_Shape_01 PEN_STYLE 'INSIDE_FRAME'
PROPERTY PageHeader_Shape_01 BRUSH_STYLE 'DIAG_CROSS'
PROPERTY PageHeader_Shape_01 PEN_STYLE 'ALTERNATE'
RETURN
-- Action: Break Footer 0 [Before Generate]
-- Break Footer Shape
PROPERTY BreakFooter_Shape_01 BRUSH_STYLE 'B_DIAGONAL'
PROPERTY BreakFooter_Shape_01 PEN_STYLE 'DASH_DOT_DOT'
RETURN
-- Action: Report Footer [Before Generate]
-- Report Footer Shape
PROPERTY ReportFooter_Shape_01 BRUSH_STYLE 'SOLID'
PROPERTY ReportFooter_Shape_01 PEN_STYLE 'DASH_DOT'
RETURN

```

For your viewing pleasure, here is a report that illustrates the use of new BRUSH_STYLE and PEN_STYLE properties of Shape control.

http://www.razzak.com/tips/InvoiceSummary_Brush_Style_Pen_Style.pdf

In addition, notice the feature of designing a custom "Cover Page" and Page Header and Page Footer options, not to mention the compact size of PDF document.

Have fun implementing BRUSH_STYLE/PEN_STYLE properties in Reports!

9.18 Using Design & Print Password Options for Reports/Labels

Your R:BASE report, based on a complex view, secure table(s), or even embedded code, can be susceptible to anyone having the access to database.

Printing a report with sensitive data or having access to modify the report layout, art work, or underlying embedded code could be surprising.

Did you know that you can now assign Design and/or Print Passwords to all of your Reports and Labels to protect your valuable code and the integrity of your Reports and Labels?

R:BASE includes the ability to set Design and Print Passwords to protect access to a Report/Label designer as well as to Print Report/Label.

While in Report/Label Designer, if assigned, R:BASE will prompt for a password the next time the Report/Label is opened in the designer or using the PRINT reportname or LBLPRINT labelname commands.

To assign Design and/or Print passwords for Reports ...

Report Designer|Main Menu|Report|Passwords|Set Design Password
 Report Designer|Main Menu|Report|Passwords|Set Print Password

To assign Design and/or Print passwords for Labels ...

Label Designer|Main Menu|Label|Passwords|Set Design Password
 Label Designer|Main Menu|Label|Passwords|Set Print Password

Note: Passwords are limited to 32 characters and are case sensitive.

9.19 Using Printer Control Codes

Did you know that you can use Printer Control Codes (PCC) in R:BASE for Windows? Using the new PCC (Printer Control Codes) Label control in either the Label or Report Designers, you can now send any Printer or POS Cash Drawer code to control the features of your printer or any other device that accepts application codes. Here's how:

1. Start R:BASE and open the Database Explorer
2. Select Reports from the Group Bar and select an existing report
3. Right click on it and select "Design Report"
4. In the Designer, notice the new "PCC Label" control under Advanced Controls
5. While in Report Designer, select the new PCC Label control and place it on the appropriate band of the report and then enter the correct code in the "Set Value" box.

Let's assume that the control code you must send to your printer to have it open your cash drawer is 27, 112, 0, 50, 250. In that particular case, you'll need to enter the following code in Set Value box to open the cash drawer before printing the receipt in R:BASE POS system.

Enter "27 112 0 50 250" into the "Set Value" box.

Another good scenario would be where the original tractor feed forms frequently require filling in data at the top of the form, which is above the area where the print head sits at rest. In order to avoid wasting the first form, the user would need back up the form to line up the top of the form with the top of the print head which would be pretty inconvenient. The other choice would be to roll up the forms on the printer and waste the first form by leaving it blank (wastes paper and is not good for the environment or trees!).

So, to achieve that goal, now all you need to do is to place the PCC label on appropriate band of the report, enter the printer specific codes, and you are done. All hardware specific codes, when placed while in the Label or Report Designer, are only enforced when printing the label or report to a printer using the **OPTION PRINTER** command.

9.20 Using Scientific Notation for Display Format

While in Report Designer, you may leave the default format of any underlying data type, or you may customize the display format of all data-aware controls as follows:

Report Designer | Select Appropriate Control | Right-Click | Display Format

For controls with numeric data types, such as, INTEGER, REAL, CURRENCY, DOUBLE, NUMERIC, and BIGNUM, notice the following available options:

#,0;-#,0
 #,0.00;-#,0.00
 #,0.00;(#,0.00)
 \$#,0.00;(\$#,0.00)

```

$#,0.00;-$#,0.00
$#,0;-$#,0
$#,0;($#,0)
0 %
0.00 %

```

You may overwrite the display format and further customize, if you wish.

Did you know that you can customize the Display Format for Scientific Notation?

Here's how ...

Use the Display Format as follows:

```
##,E+00
```

Where ...

```
## should be for the significant number
00 is the number of digits to be displayed after the E.
```

Notes:

- To not show a '+', use "##,E-00".
- A sign will only be displayed if exponent is negative.
- If any of the strings "E+", "E-", "e+", or "e-" are contained in the Display Format string, the number is formatted using scientific notation.
- A group of up to four "0" characters can immediately follow the "E+", "E-", "e+", or "e-" to determine the minimum number of digits in the exponent. The "E+" and "e+" formats cause a plus sign to be output for positive exponents and a minus sign to be output for negative exponents.
- The "E-" and "e-" formats output a sign character only for negative exponents.

9.21 Using the Persistent Space Property

Did you know that you can use the new Persistent Space Property to determine whether the space reserved for header and footer band is kept, even if the contents are not printed?

R:BASE includes two new properties for Page Header and Page Footer bands.

Persistent Space On First Page

Determines whether the space reserved for header or footer band is kept, even if the contents are not printed

Persistent Space On Last Page

Determines whether the space reserved for header or footer band is kept, even if the contents are not printed

Report Designer | Page Header Band | Right-Click | Options ...

```

[ ] Persistent Space of First Page
[ ] Persistent Space of Last Page

```

Report Designer | Page Footer Band | Right-Click | Options ...

```

[ ] Persistent Space of First Page
[ ] Persistent Space of Last Page

```

Label Designer | Page Header Band | Right-Click | Options ...

- Persistent Space of First Page
- Persistent Space of Last Page

Label Designer | Page Footer Band | Right-Click | Options ...

- Persistent Space of First Page
- Persistent Space of Last Page

By default, these settings are enabled to maintain the current properties and format of all your existing reports and labels

9.22 When to Use Report Actions vs. Report Band EEPs

Do you know when it is appropriate to use Report Actions versus using Report Band EEPs?

In the R:BASE Report/Label Designer there are now several different options now available for developers and users to run custom code within reports and labels. There are "Report Actions" and "Report Band EEPs".

With Actions, there are several levels that your custom code can be executed during a report's stages of use. The levels include:

- Before Preview - executes code before a report is previewed
- After Preview - executes code after a report is previewed
- Before Generate - executes code before a report is generated
- After Generate - executes code after a report is generated
- On Before Design - executes code before the designer is launched
- On After Design - executes code after the designer is closed

A Report Action is geared to be used for initializing variables and creating temporary or permanent tables and views. The reason for this is because Report Actions are only intended to be executed once when the report is generated in the R:BASE Report Generator.

When using Report Band EEPs, you can run custom code before or after a report band, i.e. Page Header (PH), Report Footer (RF), etc., is generated. As R:BASE reports are generated from top to bottom, the Report Band EEPs will execute in this same fashion. If you right click on any report band, you will see the following options for EEPs:

- Before Generate Custom EEP
- After Generate Custom EEP

If you intend to alter the characteristic of any report field, i.e. color, visibility, size, etc., then you must issue the appropriate PROPERTY and GETPROPERTY commands within your Report Band EEPs.

Part



10 File Gateway

10.1 Defining Column Count When Importing Data

Did you know that while automating the entire File Gateway Import routines to import MS Spreadsheet (.xls) or .CSV file into R:BASE, now you can specify the exact number of columns to import?

An additional "OPTION COLUMN_COUNT nnn" has been implemented to simplify your entire load process.

-- Example:

```
GATEWAY IMPORT filename.csv CREATE newtable OPTION COLUMN_COUNT 5
```

This command will import only first 5 columns from the CSV file.

10.2 Defining Header/Footer when Using GATEWAY EXPORT XLSW

When automating GATEWAY EXPORT XLSW options, did you know that you can also include Header and Footer text?

Two additional OPTIONS are implemented when exporting data as XLSW.

```
HEADER_TEXT value  
FOOTER_TEXT value
```

-- Example 01:

```
-- Start here  
CONNECT RRBYW17  
SET ERROR MESSAGE 2077 OFF  
DELETE InvoiceHeader.xls  
SET ERROR MESSAGE 2077 ON  
GATEWAY EXPORT XLSW InvoiceHeader.xls +  
SELECT TransID AS `Invoice No`, +  
TransDate AS `Invoice Date`, +  
NetAmount AS `Sub Total`, +  
Freight, Tax, +  
InvoiceTotal AS `Invoice Total` +  
FROM InvoiceHeader ORDER BY TransID +  
OPTION COL_NAMES ON +  
|HEADER_TEXT Customer Invoices +  
|FOOTER_TEXT Quarterly Report +  
|SHEET_NAME Invoices +  
|ACTION OPENVIEW  
RETURN  
-- End here
```

Notes:

- You can customize the column names with spaces by surrounding the names with IDQUOTE character.
- Once the worksheet is opened, notice the header information in Row 1, Cell 1 (A1), and the footer information in the last row, Cell 1.
- You may also notice the "Number Format" for Invoice No, Invoice Date, Net Amount, Freight, Tax, and Invoice Total. All data types, such as INTEGER (Number), DATE, and CURRENCY are now reflected accordingly and not formatted as custom.

- You may also customize and automate the file extension (.xls or .xlsx) by using the PLUGIN RRegistry Command.

10.3 Import XLS/XLSX Spreadsheet Column Mapping

Did you know that you can map R:BASE columns to XLS/XLSX spreadsheet columns in order to control the data import into specific locations?

You can include the ADD_MAPPING parameter in the GATEWAY command syntax to map an R:BASE column to the spreadsheet column.

Here's how:

-- This command will only append Cell B, Cell C and Cell D data into the Company, CustAddress and CustState columns of the Customer table from the Customer.xls file.

```
GATEWAY IMPORT XLS Customer.XLS APPEND Customer +
OPTION ADD_MAPPING Company=B|ADD_MAPPING CustAddress=C|ADD_MAPPING
CustState=D
```

10.4 Multi-Tab Excel Workbook with Headers Using GATEWAY EXPORT XLSW

Here's an example to create multi-tab Excel Workbook with Headers using GATEWAY EXPORT XLSW options,

```
-- GATEWAY_Multi_Tab_EXPORT_XLS_XLSW.RMD
-- Author: A. Razzak Memon
-- Last Updated:
IF (CVAL('DATABASE')) <> 'RRBYW19' OR (CVAL('DATABASE')) IS NULL THEN
    CONNECT RRBYW19 IDENTIFIED BY NONE
ENDIF
CLEAR VARIABLES vCaption,vPause3Message,vPause4Message,vCheckExtension
SET VAR vCaption TEXT = 'Multi-Tab XLS/XLSX Workbook'
SET VARIABLE vPause3Message TEXT = NULL
SET VARIABLE vPause4Message TEXT = NULL
SET VARIABLE vCheckExtension TEXT = NULL
SET VARIABLE vFileName TEXT = NULL
SET VAR vPause3Message = +
    ((CHAR(013))+ ' Now Preparing Multi-Tab XLS/XLSX Workbook')
CLS
PAUSE 3 USING .vPause3Message +
    CAPTION .vCaption +
    ICON INFO +
    OPTION MESSAGE_FONT_NAME Tahoma +
    |MESSAGE_FONT_COLOR GREEN +
    |MESSAGE_FONT_SIZE 11 +
    |THEMENAME Vista CG
-- Delete previously created RRBYW19_Multi_Tab_WorkBook.xl??, if any
SET ERROR MESSAGE 2077 OFF
DELETE RRBYW19_Multi_Tab_WorkBook.xl??
SET ERROR MESSAGE 2077 ON
-- Define file name with appropriate file extension (.xls or .xlsx)
-- .xlsx for MS Office Excel 2007 or higher
PLUGIN RRegistry 'vCheckExtension|CHECK_KEY|HKCR|.xlsx|'
```

```
IF vCheckExtension = 'TRUE' THEN
    SET VARIABLE vFileName = 'RRBYW19_Multi_Tab_WorkBook.xlsx'
ELSE
    SET VARIABLE vFileName = 'RRBYW19_Multi_Tab_WorkBook.xls'
ENDIF
-- Create Tab 1 (Customers)
SET VAR vPause4Message = +
    ((CHAR(013))+ ' Now Adding Tab 1 - Customers')
PAUSE 4 USING .vPause4Message +
    CAPTION .vCaption +
    ICON INFO +
    OPTION MESSAGE_FONT_NAME Tahoma +
    |MESSAGE_FONT_COLOR GREEN +
    |MESSAGE_FONT_SIZE 11 +
    |THEMENAME Vista CG
GATEWAY EXPORT XLSW .vFileName +
    SELECT CustID AS `Cust ID`, +
    Company AS `Company Name`, +
    CustAddress AS `Address`, +
    (CustCity+', '&CustState&CustZip) AS `City, State and Zip Code`, +
    CustPhone AS `Phone Number` +
    FROM Customer ORDER BY Company +
    OPTION COL_NAMES ON +
    |HEADER_TEXT Customers +
    |SHEET_NAME Customers|SHEET_INDEX 1
-- Create Tab 2 (Invoice Headers)
SET VAR vPause4Message = +
    ((CHAR(013))+ ' Now Adding Tab 2 - Invoice Headers')
PAUSE 4 USING .vPause4Message +
    CAPTION .vCaption +
    ICON INFO +
    OPTION MESSAGE_FONT_NAME Tahoma +
    |MESSAGE_FONT_COLOR GREEN +
    |MESSAGE_FONT_SIZE 11 +
    |THEMENAME Vista CG
GATEWAY EXPORT XLSW .vFileName +
    SELECT TransID AS `Invoice No`, +
    CustID AS `Cust ID`, +
    TransDate AS `Invoice Date`, +
    NetAmount AS `Sub Total`, +
    Freight, Tax, +
    InvoiceTotal AS `Invoice Total` +
    FROM InvoiceHeader ORDER BY TransID +
    OPTION COL_NAMES ON +
    |HEADER_TEXT Invoices +
    |SHEET_NAME Invoices|SHEET_INDEX 2 +
    |ACTION OPENVIEW
CLEAR VARIABLES vCaption, vPause%, vCheckExtension, vFileName
CLS
RETURN
```

Notes:

- Notice the use of PLUGIN RRegistry Command to customize and automate the file name extension (.xls)

or .xlsx).

- You can customize the column names with spaces by surrounding the names with IDQUOTE character.
- Once the worksheet is opened, notice the header information in Row 1, Cell 1 (A1), and the footer information in the last row, Cell 1.
- You may also notice the "Number Format" for Invoice No, Invoice Date, Net Amount, Freight, Tax, and Invoice Total. All data types, such as INTEGER (Number), DATE, and CURRENCY are now reflected accordingly and not formatted as custom.

10.5 Suppressing GATEWAY IMPORT Messages

The R:BASE File Gateway utility contains many powerful features to allow the importing and exporting of data. You may use the GUI interface to import data by clicking on the File Gateway icon on the main R:BASE toolbar. Once the File Gateway module initializes, select the Import tab. On the Import tab choose to either add the data to an "Existing Table", Create a "New Table" or TEMPORARY table.

Using the GATEWAY IMPORT command in a command file, you can automate the entire task of importing desired data formats from your end-users and/or clients. There are NO limits on importing numerous records into R:BASE databases with the GATEWAY command.

While automating the import process, be aware of both your source file structure and table structure for the command to function properly. The table and file structure MUST coincide with each other.

While importing the data, GATEWAY Import utility provides an animated status progress windows as well as -ERROR- message(s), if any.

Did you know that you can suppress both PROGRESS and MESSAGES while importing the data into an R:BASE table?

You have two additional OPTIONS for GATEWAY IMPORT syntax as follows:

SHOW_PROGRESS

Indicates whether to display status dialog during the import process. Set SHOW_PROGRESS to OFF if you wish to hide the animated status box during the import process. The default is set to ON.

MESSAGES

If ON, during import process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire import process will be in "silent" mode. The default is set to ON.

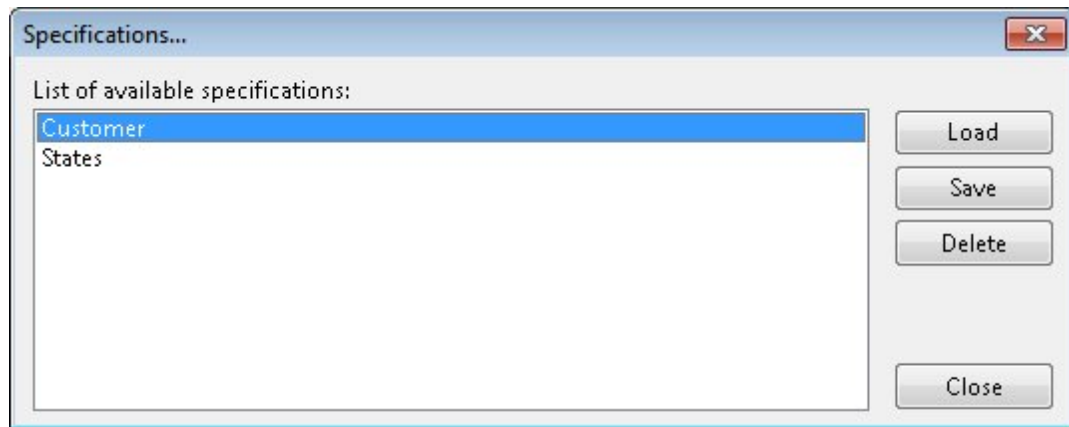
For complete details and examples, enter HELP IMPORT at the R> Prompt.

10.6 Using File Gateway Specification Files

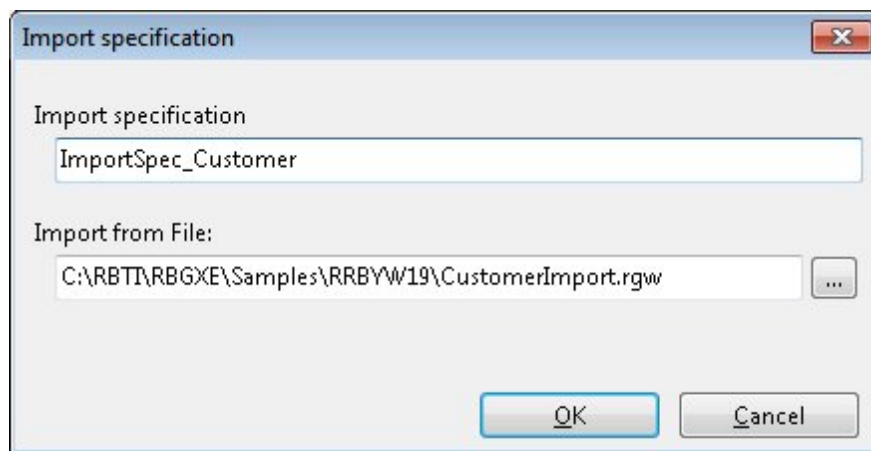
When using the File Gateway utility to import data into an existing table or when using the Export Wizard, specification files (.rgw) can be used to save your import/export options and settings to load from a previous data transfer.

As part of the latest update, it is now easier to manage multiple specification files in the same working folder with an enhanced "Save As" button.

During the import/export process, the "Specifications..." button is available. The specification file will store your import options for type, source file, fields, settings and mode.



The specification files can be loaded for the current data transfer and saved for future transfers. The specification files are stored in the current folder with the .RGW file extension.



Specification files can be utilized using both the File Gateway user interface and the GATEWAY.

GATEWAY Import Example:

```
GATEWAY IMPORT XLS Customer_New.xls APPEND Customer +
OPTION SHEET_INDEX 1 +
| SPECIFICATION_FILE_NAME Customer_Import.rgw +
| FIRST_ROW 2 +
| SHOW_PROGRESS ON +
| MESSAGES OFF
```

GATEWAY Export Example:

```
GATEWAY EXPORT CSV Customer_Release.csv +
SELECT Company,CustAddress,CustCity,CustState,CustZip FROM Customer +
OPTION COL_NAMES OFF +
| SHOW_PROGRESS ON +
| MESSAGES OFF +
| SPECIFICATION_FILE_NAME Customer_Release.rgw
```

Part

XI

11 R:BASE Editor

11.1 Commenting and Uncommenting Commands

There are several different methods you can use in order to comment and uncomment lines or blocks of code in your command files.

To comment an individual line, add two hyphen characters "--" to the beginning of the line. In R:BASE Editor, the syntax highlighting will alter the display and change the font color to pink and the style to italicized. In the following example;

```
CLEAR VAR vResult
```

the command will become:

```
--CLEAR VAR vResult
```

Another use of characters that will comment your code is the asterick character preceding a set of parentheses "*()", with the desired commented text or commands enclosed within the parentheses. In the following example;

```
PLUGIN RPDFMerge 'vResult +  
|ACTION MERGE +  
|DOC_LIST_FILE PDFFilesToMerge.LST +  
|SHOW_SETUP_DIALOG ON +  
|OUTPUT_FILE OneBigMergedFile.PDF '
```

the commands will become:

```
*(  
  PLUGIN RPDFMerge 'vResult +  
  |ACTION MERGE +  
  |DOC_LIST_FILE PDFFilesToMerge.LST +  
  |SHOW_SETUP_DIALOG ON +  
  |OUTPUT_FILE OneBigMergedFile.PDF '  
)
```

Although this option is still supported in R:BASE, it is now recommended that you use an alternative method to avoid confusing parentheses with your R:BASE expressions. The suggested alternative is the use of a set of squiggly brackets "{}", with the desired commented text or commands enclosed within the squiggly brackets . In the following example;

```
PLUGIN RPDFMerge 'vResult +  
|ACTION MERGE +  
|DOC_LIST_FILE PDFFilesToMerge.LST +  
|SHOW_SETUP_DIALOG ON +  
|OUTPUT_FILE OneBigMergedFile.PDF '
```

the commands will become:

```
{  
  PLUGIN RPDFMerge 'vResult +  
  |ACTION MERGE +  
  |DOC_LIST_FILE PDFFilesToMerge.LST +  
  |SHOW_SETUP_DIALOG ON +  
}
```

```
/OUTPUT_FILE OneBigMergedFile.PDF '  
}
```

Other methods to comment an entire block of text or code include the options in the R:BASE Editor:

Highlight the entire block of text

Right Click > Block > Comment

You can uncomment the entire block of your code in the R:BASE Editor by:

Highlight the entire block of text

Right Click > Block > Uncomment

You may also want to explore other options in completely re-designed R:BASE Editor!

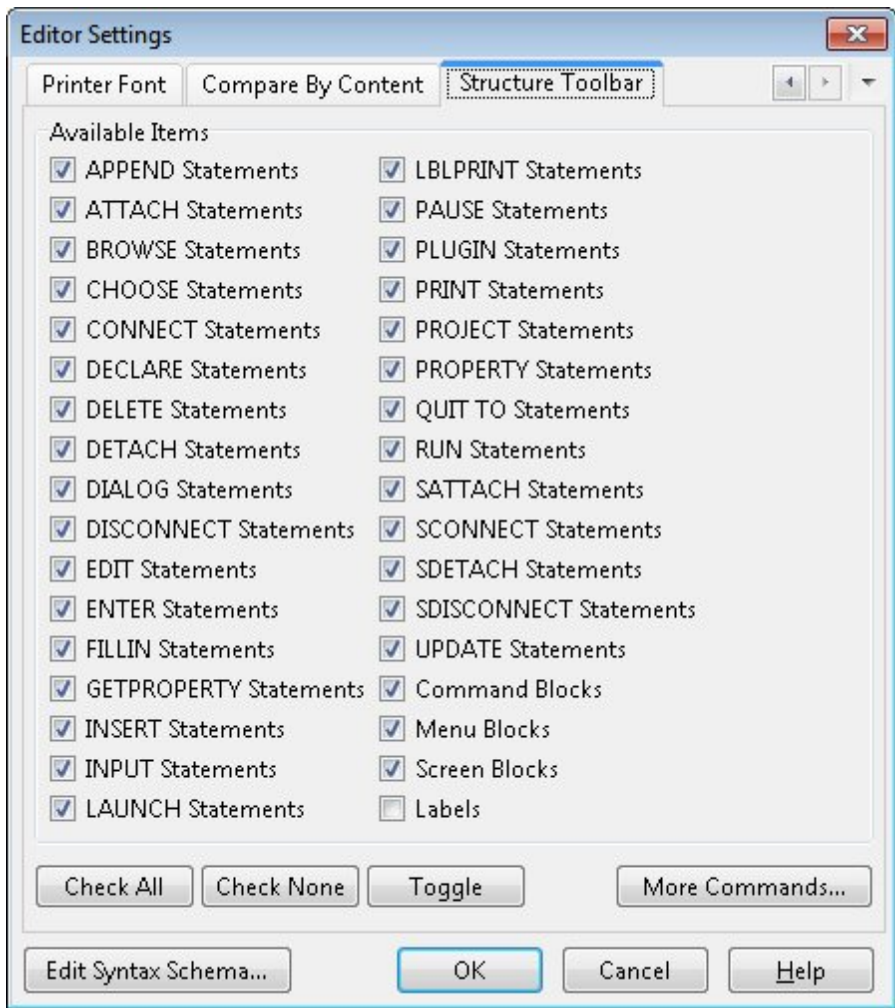
11.2 Customizing RBEEdit Structure Toolbar Options

Did you know that you can customize the R:BASE Editor (RBEEdit) Structure Toolbar settings to include custom structure options when editing or analyzing your application or command file?

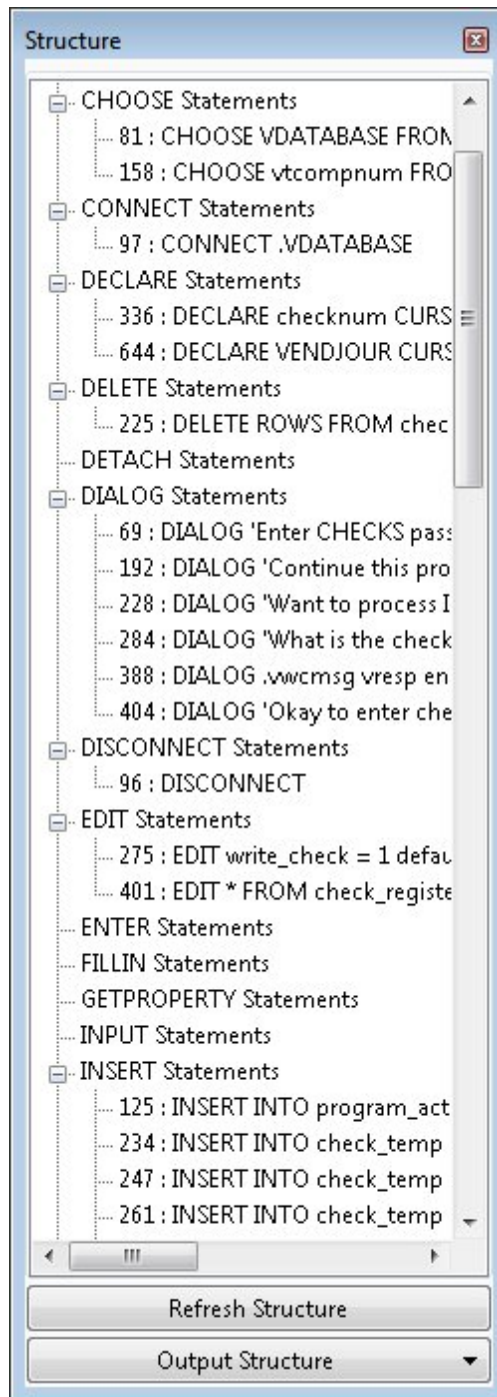
Here's how:

From the main Menu Bar, select > Settings > R:BASE Editor. Then, select the "Structure Toolbar" tab.

Once you check all appropriate options accordingly, click on OK button to save your custom settings.



Now when editing your application or command file, clicking on the [Refresh Structure] with document all selected options.



Clicking on [Output Structure] button will let you save the entire structure with all details as follows:

- Printer
- Text File
- PDF File
- Copy to Clipboard

Have fun taking advantage of Expanded Structure as well as Output options in R:BASE eXtreme.

11.3 Integrating the External R:BASE Editor

With the added power and features of R:BASE Editor, <http://www.rbedit.com/>, if you wish to integrate the external editor into R:BASE, all you have to do is to simply use the following command at the R> prompt:

```
SET EDITOR 'C:\RBTI\RBEEdit\RBEEdit.exe'
```

This will set the default editor to your choice of R:BASE Editor.

The following will reveal the setting for your default editor.

```
SHOW EDITOR
```

The default editor setting can also be captured in a variable using the following Function:

```
(CVAL('EDITOR'))
```

You may also use the following option in your R:BASE configuration file (line #86):

```
EDITOR C:\RBTI\RBEEdit\RBEEdit.exe
```

11.4 New File Encoding Options

Did you know the R:BASE Editor was recently enhanced to set/preserve the file encoding (ANSI, UTF-8, Unicode) of loaded command files?

From the main menu bar, select "File" > "Encoding"

Also when selecting to save a new command file, an "Encoding" option is available within the "Save As" dialog to select the file encoding method.

Supported Versions:

- R:BASE X (Build...: 10.0.2.11130 or higher)

11.5 Running Your Code Within RBEEdit

Did you know that now you can RUN your code while creating new code or updating existing code while in RBEEdit?

Here's how:

1. Start R:BASE
2. Open your existing command file or simply create a new one and use the following code:

```
CLS
SET VAR vTest TEXT = 'R:BASE Rocks!'
DIALOG 'Enter Text Messge' vTest=26 vEndKey 1 +
CAPTION 'Using AUTOSELECT' ICON APP +
OPTION AUTOSELECT TRUE +
|THEMENAME R:BASE Rocks!
RETURN
```

3. While still in RBEEdit, either click on the new Run [>] button on the tool bar or use [Ctrl + F9] keys to RUN your code.

If you need to make any changes, do it while in RBEEdit and test your code again without closing RBEEdit.

11.6 Wrapping Commands Using the Right Margin

Did you know that you can customize the location of the right margin in R:BASE Editor, in order expand the display of command files and EEPs?

The Right Margin is helpful in automatically displaying and wrapping long commands within the defined screen, when "Word Wrap" and "Break Word at Right Margin" are both enabled.

Here's how:

01. Within the R:BASE Editor, select "Editor Settings" > "General" tab, and locate the "Break Word at Right Margin" and "Word Wrap" check boxes in the right column.
02. Make sure both settings are enabled by clicking on the check box if it is empty.
03. While the "Editor Settings" is still displayed, select the "Display" tab and locate the "Right Margin" field (default 80 characters).
04. Using the available arrow buttons or by typing the new value, increase the value anywhere from 100 to 125, based upon your available screen space on the right.
05. Click the "OK" button.

Where long R:BASE commands are listed, you will see the extra space become populated, allowing for more of the command files to be displayed on the screen.

Have fun customizing the many options available within the R:BASE Editor!

Part

XII

12 Trace Debugger

12.1 Accessing Database Settings While in TRACE Mode

Did you know that you can access current database settings while in TRACE mode?

The Data Dictionary, [F3] key, includes an additional tab to display the current Database Settings, which is very useful when tracing code.

Here's a comprehensive list of items/features when pressing the [F3] key:

- Databases (Name, Date/Time, Size)
- Tables (Name, Comment) *
- Columns (Table/View Name, Name, Comment) *
- Key/Indexes (ID, Type, Table Name, Reference) *
- Variables (Name, Data Type, Value)
- Views (Name, Comment) *
- Forms (Name, Table, Comment) *
- Reports (Name, Table, Comment) *
- Labels (Name, Table, Comment) *
- Functions (Name, Category, Syntax)
- Stored Procedures (Name, Comment) *
- System Tables (Name) *
- System Columns (Table/View Name, Name, Comment) *
- Themes (Name)
- Printers (Name)
- External Forms (External Form Name, Date/Time, Size)
- Applications (Application Name, Date/Time, Size)
- Colors (Color, Name, Integer Value, RGB Value)
- Control Types (Name, Comment)
- Database Settings (Parameter, Value, Description) **

* Only visible when connected to database

** If not connected to database, default R:BASE settings are shown.

12.2 Clearing Error List

In the Trace Debugger, an Error List Toolbar is available to display programming errors that occur during code debugging. If the Error List becomes too populated, there was no method to clear the list without restarting R:BASE.

Now, a new menu option is available to clear the error list.

From the Menu Bar, choose "Errors" > "Clear Error List"

12.3 Managing Watch Variables in TRACE Mode

While tracing R:BASE code, using the TRACE command, you may press the [F6] key to Add/Remove Watch Variable(s) or simply right-click on the watch variables window to bring up the speed menu option, and then select click on the Add/Remove Watch Variable(s).

Using the Add/Remove Variable(s) console window, typically, you would select/highlight the variable from the list of Available Variables and then click on the right arrow [>] button to add.

Did you know that you can also Add/Remove variable(s) by using the double-click?

Part

XIII

13 BLOB Editor

13.1 Customizing the BLOB Editor

Did you know that now you can customize the caption and icon of R:BASE BLOB Editor?

One of the more exciting features in R:BASE for Windows is the ability to add, edit, delete, and display binary files-Binary Large Objects, or BLOBs. This term refers to binary data as opposed to ASCII data. The data types VARBIT and BITNOTE were added for storing binary files within an R:BASE database. In addition, the VARCHAR data type lets you store large ASCII data files (Large Objects, or LOBs). The data for VARBIT and VARCHAR data types is stored in the fourth R:BASE data file, DBName.RB4 or DBName.RX4.

In addition, you may also use R:BASE BLOB Editor to create a new or update an existing files, such as RTF, BMP, TXT, JPG, GIF, PNG, and more...

While deploying R:BASE applications with integrated features of R:BASE BLOB Editor, now you can customize the caption and icon. Here's how:

First, use the following PROPERTY commands to customize R:BASE BLOB Editor.

```
PROPERTY BLOB_EDITOR CAPTION ' This is the R:BASE you have always wanted!'  
PROPERTY BLOB_EDITOR ICON 'IntoTheFuture.ico'
```

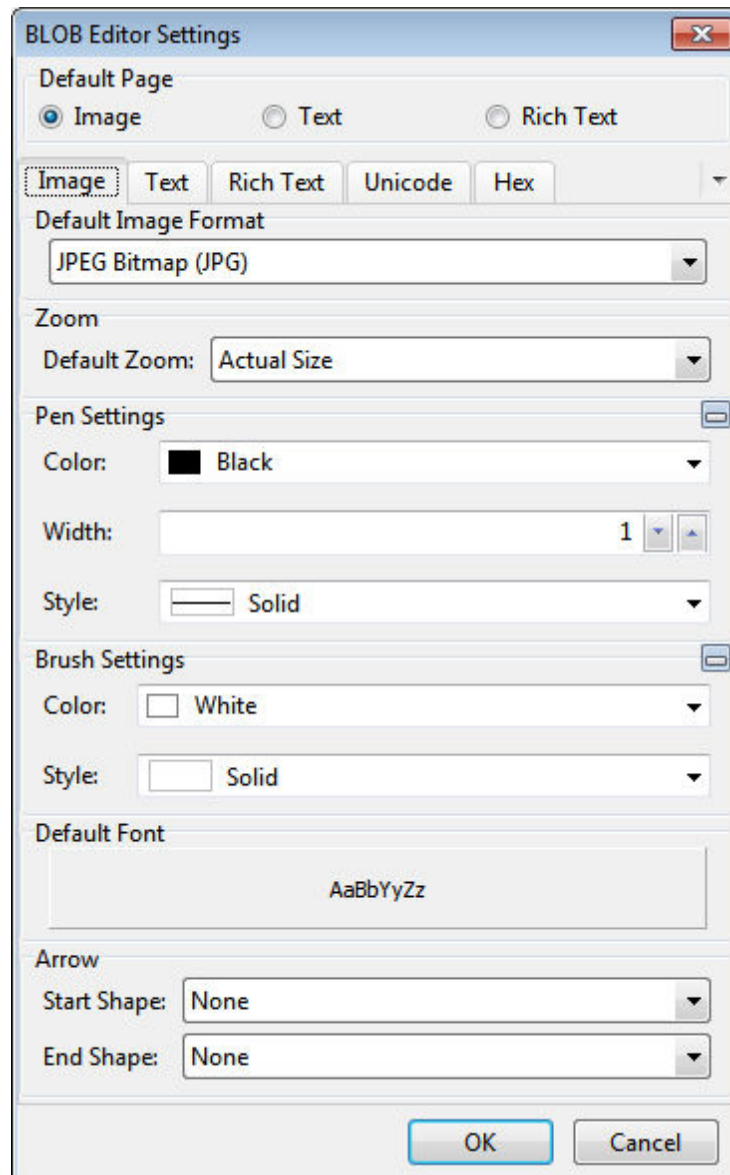
Then, use the following command to create a new Rich Text File with your favorite logo into a single table column.

```
RBBE NewRTFFile.RTF
```

13.2 Defining Default Settings for R:BASE BLOB Editor

Did you know that now you can define R:BASE BLOB Editor Default Settings? Here's how:

01. Start R:BASE
02. Database Explorer | Main Menu | Settings | BLOB Editor ...



Part

XIV

14 Commands

14.1 Assigning Line Numbers using the SELECT Command

Did you know that you can assign the line numbers for the SELECT command?

Use the (CVAL('ROWCOUNT')) function to achieve that goal.

Here's how:

1. Start R:BASE
2. CONNECT ConComp
3. At the R> Prompt:

```
SELECT (INT(CVAL('ROWCOUNT'))) AS LineNo,Company=35, +
(CustCity+', '&CustState&CustZip)=35 AS CityStateZip FROM Customer
```

Notice the line counter!

4. If you would like to print that select statement output as PDF, use the following example:

```
SET VAR vLines = (INT(CVAL('LINES')))
SET LINES 0
OUTPUT CustomerList.PDF PDF
SELECT (INT(CVAL('ROWCOUNT'))) AS LineNo,Company=35, +
(CustCity+', '&CustState&CustZip)=35 AS CityStateZip FROM Customer
OUTPUT SCREEN
SET LINES .vLines
```

That will send the output as CustomerList.PDF using the "built-in" PDF capabilities of R:BASE OUTPUT command.

Use the LAUNCH CustomerList.PDF command to view the results.

14.2 Attaching Server Tables as TEMPORARY Tables

Did you know that now you can attach server tables as TEMPORARY tables?

Among so many enhancements and features in R:BASE, the traditional SATTACH command has been enhanced to include the option for attaching server tables as TEMPORARY.

Syntax:

```
SATTACH TEMPORARY tablename AS tablealias USING ...
```

The optional TEMPORARY parameter allows you to create a Temporary Table with the SATTACH command and all supported parameters. The temporary tables will disappear when the database is DISConnected.

Use this option to take the exact snapshot of the sever table to manage and analyze the data as you wish.

Note:

Any changes made to the temporary table will not be reflected upon the original SQL data source. The sole purpose of this enhancement is to let R:BASE attach server tables as TEMPORARY tables specific to that User/R:BASE session even when the database is connected with STATICDB ON.

14.3 Automatic Resizing for Custom Images

Did you know that the DIALOG command has been enhanced with the RESIZE_ICON parameter, to resize a specified image to 32x32 pixels?

When using the ICON_FILE parameter, a custom image can be added to the DIALOG window to replace one of the available icons R:BASE provides.

If the custom image is smaller or large than 32x32 pixels, the display of the image will not fit precisely within the image holder.

The RESIZE_ICON (ON/OFF) parameter will resize the image (except .ico) to 32x32.

Example:

```
DIALOG 'Do you wish to continue?' vResponse vEndKey YES +
CAPTION 'Confirm' OPTION ICON_FILE Logo128x128.jpg +
|RESIZE_ICON ON +
|MESSAGE_FONT_COLOR NAVY +
|MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_SIZE 10 +
|MESSAGE_BOLD ON
```

NOTE: The following image formats are supported. There is no size limitation, though 32x32 is recommended.

- wmf | Windows Metafiles
- emf | Enhanced Metafiles
- ico | Icon Files
- bmp | Windows Bitmap Files
- png | Portable Network Graphics
- jpeg | JPEG Image File
- jpg | JPEG Image File
- tif | TIFF Image File
- tiff | TIFF Image File
- gif | Compuserve GIF Image
- jp2 | JPEG 2000 Image
- j2k | JPEG 2000 Image
- jpc | JPEG 2000 Image
- tga | Truevision Targa File
- vst | Truevision Vista File
- afi | Truevision Targa File
- pcx | ZSoft PC Paintbrush
- pcc | ZSoft PC Paintbrush
- dcx | DCX Image File

14.4 Capturing Screenshots to the Windows Clipboard

Did you know that you can capture a screen shot of a running form, copy it to the Windows Clipboard, and then paste the image for your internal documentation, support e-mail (with built-in tools), or even save as a snapshot of given date and time?

As the current PROPERTY command supports capturing an image of the current window or screen and places the content in a file, the command can now also place the image content in the clipboard.

-- To capture the window to the clipboard

```
PROPERTY RBASE_FORM CAPTURE_FROM_WINDOW 'CLIPBOARD'
RETURN
```

-- to capture part of the screen to the clipboard

```
PROPERTY RBASE_FORM 'CAPTURE_FROM_SCREEN|23, 24, 934, 800' 'CLIPBOARD'
RETURN
```

14.5 Changing CASCADE Options

With prior versions/builds of R:BASE, if you have already added the CASCADE option to a table with a Primary Key (PK) that automatically executes the UPDATE and DELETE options, and if you wish to change the default option to UPDATE only, use the following command at the R> prompt:

```
ALTER TABLE <PKtablename> ADD CASCADE UPDATE
```

This will change the CASCADE option to UPDATE only.

If you wish to change the default option to DELETE only, use the following command at the R> prompt:

```
ALTER TABLE <PKtablename> ADD CASCADE DELETE
```

If you wish to change the default option to both, use the following command at the R> prompt:

```
ALTER TABLE <PKtablename> ADD CASCADE
```

And finally, if you wish to cancel the CASCADE option altogether, use the following command at the R> prompt:

```
ALTER TABLE <PKtablename> DROP CASCADE
```

You may also implement these options using the enhanced R:BASE Data Designer (RBDEFine).

14.6 Changing Report Margins Using the PRINT Command

Did you know that you can change the report margins dynamically using the enhanced PRINT command with the "margin" parameters?

Here's how:

Four additional PRINT OPTION parameters are now supported to change Left, Top, Right and Bottom margins of report.

Syntax:

```
PRINT reportname OPTION outputvalue|MARGIN_xxxx nnn
```

Where:

outputvalue is the parameter such as SCREEN, PRINTER, BMP, EMF, ETXT, GIF, HTML, JPG, PDF, RTF, TIFF, TXT, WMF, XHTML, or XLS.

MARGIN_xxxx are the parameters for:

- MARGIN_LEFT
- MARGIN_TOP
- MARGIN_RIGHT
- MARGIN_BOTTOM

nnn is the REAL data type value; such as 0.5, 1.0, or 0.75

MARGIN_LEFT - specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).

MARGIN_TOP - specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0,

then that component will print 0.25 inches from the edge of page (or at the top margin).

MARGIN_RIGHT - specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.

MARGIN_BOTTOM - specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.

```
-- Example 01:
PRINT CustomerList OPTION PRINTER|MARGIN_LEFT 1.25

-- Example 02:
PRINT CustomerList OPTION PRINTER|MARGIN_TOP 0.50

-- Example 03:
PRINT CustomerList OPTION PRINTER|MARGIN_RIGHT 0.75

-- Example 04:
PRINT CustomerList OPTION PRINTER|MARGIN_BOTTOM 0.50

-- Example 05:
PRINT CustomerList OPTION PRINTER +
|MARGIN_LEFT 1.25 +
|MARGIN_TOP 0.50 +
|MARGIN_RIGHT 0.75 +
|MARGIN_BOTTOM 0.50
```

14.7 Clearing Stubborn Table Locks

In a multi-user environment, have you ever been in a situation where a table is left locked and you have no way to clear the lock(s), even knowing that there is no one out there using the same database?

Well, now you have the option to clear those stubborn table locks by simply using the *new* CLEAR TABLE LOCKS command.

Here's how:

```
DISCONNECT
SET MULTI OFF
CONNECT dbname
CLEAR TABLE LOCKS
```

This will CLEAR all table locks.

14.8 Creating/Editing RTF Files Outside the Database

Did you know that you can create and/or edit rich text file(s) outside the R:BASE database?

Use the new RBBEDIT command at the R> Prompt, in a Command File or in an EEP.

Example:

```
RBBEDIT ThePowerOfRBASE.RTF
```

or

RBBE ThePowerOfRB.RTF

14.9 Creating Secure and Encrypted Files

When providing any information to your clients, providers, or government agencies as R:BASE output file or a PDF report that include sensitive data, such as credit card numbers, social security numbers, date of birth, and personal client data, it is very important to provide the data as securely as you can.

When creating automated database backups using the UNLOAD ALL command, it is also important to use R:BASE encryption for any databases with owner name and passwords.

Did you know that you can create a secure output or secure PDF file using the built-in R:BASE encryption?

The latest versions of R:BASE (for Windows) include an added ENCRYPT parameter that can be used to protect and encrypt any output file.

A 512-bit encryption method is used to obscure any output data, making it unreadable without R:BASE and the decryption password.

An in-depth explanation of encryption is beyond the scope of this tip.

Simply put, encryption is a technique to prevent access to sensitive data by replacing the sensitive plain text data with illegible cipher text. The plain text is altered by built-in R:BASE encryption to make it unreadable, creating the cipher text.

Syntax:

```
OUTPUT filename.ext ENCRYPT password
-- whatever
OUTPUT SCREEN
```

R:BASE encryption password is limited to 32 characters, and is case sensitive.

Example 01: (Creating Encrypted Output File)

```
CONNECT RRBYW19
OUTPUT RRBYW19BKP.all ENCRYPT whatever
UNLOAD ALL
OUTPUT SCREEN
```

Editing or running the R:BASE encrypted output file will prompt you for the file password.

You may RUN the encrypted file with added password parameter.

Example:

```
RUN filename.ext password
Example 02: (Creating Encrypted PDF File)
CONNECT RRBYW19
PRINT SalesSumAndSubTot +
OPTION PDF +
|FILENAME SalesSumAndSubTotals.PDF +
|ENCRYPT_FILE ON +
|USER_PASSWORD RRBYW19 +
|OWNER_PASSWORD RRBYW19Owner +
|ENCRYPTION_LEVEL 128BIT
RETURN
```

For detailed syntax, enter HELP PRINT at the R> Prompt.

Opening the R:BASE encrypted PDF file will prompt you for the file password.

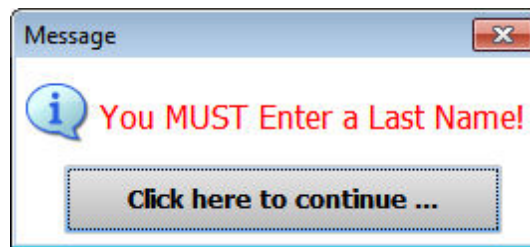
14.10 Customizing Buttons for PAUSE and DIALOG Commands

When designing and deploying R:BASE applications for mobile tablet PCs and high resolution devices, it is sometimes hard to read the text and to click on the PAUSE buttons for customized YES/NO buttons for DIALOG command.

Did you know that now you can customize the button text font for PAUSE and DIALOG commands?

You can customize the "font size", "button width" and "button height" of PAUSE button. Similarly, you can also customize the YES/NO button properties for the DIALOG buttons.

```
-- Example 01 (PAUSE with Custom Button)
CLS
PAUSE 2 USING 'You MUST Enter Last Name!' +
CAPTION 'Message' +
ICON INFO +
BUTTON 'Click here to continue ...' +
OPTION BACK_COLOR WHITE +
|MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_COLOR RED +
|MESSAGE_FONT_SIZE 12 +
|BUTTON_COLOR -16777216 +
|BUTTON_FONT_NAME Tahoma +
|BUTTON_FONT_COLOR -16777208 +
|BUTTON_FONT_SIZE 10 +
|BUTTON_FONT_BOLD ON +
|BUTTON_WIDTH 220 +
|BUTTON_HEIGHT 35
RETURN
```

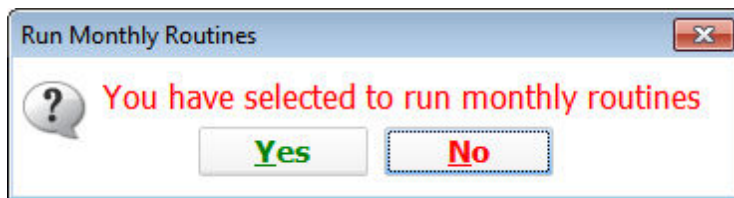


```
-- Example 02 (DIALOG with Custom Buttons)
SET VAR vCaption TEXT = 'Run Monthly Routines'
SET VAR vYesNo TEXT = NULL
SET VAR vEndKey TEXT = NULL
CLS
DIALOG 'You have selected to run monthly routines' vYesNo vEndKey No +
CAPTION .vCaption +
ICON QUESTION +
OPTION WINDOW_BACK_COLOR WHITE +
|MESSAGE_BACK_COLOR WHITE +
|MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_COLOR RED +
|MESSAGE_FONT_SIZE 12 +
|BUTTON_YES_CAPTION &Yes +
```

```

|BUTTON_YES_COLOR WHITE +
|BUTTON_YES_FONT_COLOR GREEN +
|BUTTON_YES_FONT_NAME Tahoma +
|BUTTON_YES_FONT_SIZE 11 +
|BUTTON_YES_FONT_BOLD ON +
|BUTTON_YES_WIDTH 85 +
|BUTTON_NO_CAPTION &No +
|BUTTON_NO_COLOR WHITE +
|BUTTON_NO_FONT_COLOR RED +
|BUTTON_NO_FONT_NAME Tahoma +
|BUTTON_NO_FONT_SIZE 11 +
|BUTTON_NO_FONT_BOLD ON +
|BUTTON_NO_WIDTH 85
IF vYesNo = 'No' OR vEndKey = '[Esc]' THEN
GOTO Done
ENDIF
-- Do what you have to do here ...
LABEL Done
CLS
CLEAR VARIABLES vCaption,vYesNo,vEndKey
RETURN

```



14.11 Customizing DIALOG POPUP Colors

Did you know that you can customize the DIALOG POPUP colors to match your custom application theme?

Here's how:

Example 01:

```

-- Start
-- DIALOG_POPUP_CustomColors_Razzmatazz.RMD
-- Theme: Razzmatazz
IF (CVAL('DATABASE')) <> 'RRBYW19' OR (CVAL('DATABASE')) IS NULL THEN
CONNECT RRBYW19 IDENTIFIED BY NONE
ENDIF
CLEAR VARIABLES vCustIDTxt,vCustID,vEndKey
SET VAR vCustIDTxt TEXT = NULL
SET VAR vCustID INTEGER = NULL
SET VAR vEndKey TEXT = NULL
CLS
DIALOG 'Enter a Customer ID' vCustIDTxt=8 vEndkey 1 +
CAPTION 'DIALOG Custom Colors' +
OPTION THEMENAME Razzmatazz +
|POPUP_ENABLED TRUE +
|POPUP_DIALOG_TYPE LOOKUP +

```

```

|DIALOG_EDIT_HINT Double-click for a List +
|POPUP_CAPTION CustID Selection +
|POPUP_TITLE_TEXT Select a Customer +
|POPUP_TABLE Customer +
|POPUP_SELECT Company +
|POPUP_WHERE ORDER BY Company +
|POPUP_RETURN_COLUMN CustID +
|POPUP_WINDOW_BACK_COLOR [R218,G228,B246] +
|POPUP_LIST_BACK_COLOR [R218,G228,B246] +
|POPUP_LIST_FONT_COLOR NAVY +
|POPUP_LIST_FONT_NAME ARIAL +
|POPUP_LIST_FONT_SIZE 10 +
|POPUP_LIST_FONT_BOLD ON +
|POPUP_TITLE_BACK_COLOR [R218,G228,B246] +
|POPUP_TITLE_FONT_COLOR NAVY +
|POPUP_TITLE_FONT_NAME ARIAL +
|POPUP_TITLE_FONT_SIZE 10 +
|POPUP_TITLE_FONT_BOLD ON +
|POPUP_LINES 16
-- Do what you have to do here ...
RETURN
-- End

```

Example 02:

```

-- Start
-- DIALOG_POPUP_CustomColors_RBASERocks.RMD
-- Theme: R:BASE Rocks!
IF (CVAL('DATABASE')) <> 'RRBYW19' OR (CVAL('DATABASE')) IS NULL THEN
  CONNECT RRBYW19 IDENTIFIED BY NONE
ENDIF
CLEAR VARIABLES vCustIDTxt,vCustID,vEndKey
SET VAR vCustIDTxt TEXT = NULL
SET VAR vCustID INTEGER = NULL
SET VAR vEndKey TEXT = NULL
CLS
DIALOG 'Enter a Customer ID' vCustIDTxt=8 vEndkey 1 +
  CAPTION 'DIALOG Custom Colors' +
  OPTION THEMENAME R:BASE Rocks! +
  |POPUP_ENABLED TRUE +
  |POPUP_DIALOG_TYPE LOOKUP +
  |DIALOG_EDIT_HINT Double-click for a List +
  |POPUP_CAPTION CustID Selection +
  |POPUP_TITLE_TEXT Select a Customer +
  |POPUP_TABLE Customer +
  |POPUP_SELECT Company +
  |POPUP_WHERE ORDER BY Company +
  |POPUP_RETURN_COLUMN CustID +
  |POPUP_WINDOW_BACK_COLOR 2708091 +
  |POPUP_LIST_BACK_COLOR 2708091 +
  |POPUP_LIST_FONT_COLOR WHITE +
  |POPUP_LIST_FONT_NAME ARIAL +
  |POPUP_LIST_FONT_SIZE 10 +
  |POPUP_LIST_FONT_BOLD ON +

```

```
| POPUP_TITLE_BACK_COLOR 2708091 +
| POPUP_TITLE_FONT_COLOR WHITE +
| POPUP_TITLE_FONT_NAME ARIAL +
| POPUP_TITLE_FONT_SIZE 10 +
| POPUP_TITLE_FONT_BOLD ON +
| POPUP_LINES 16
-- Do what you have to do here ...
RETURN
-- End
```

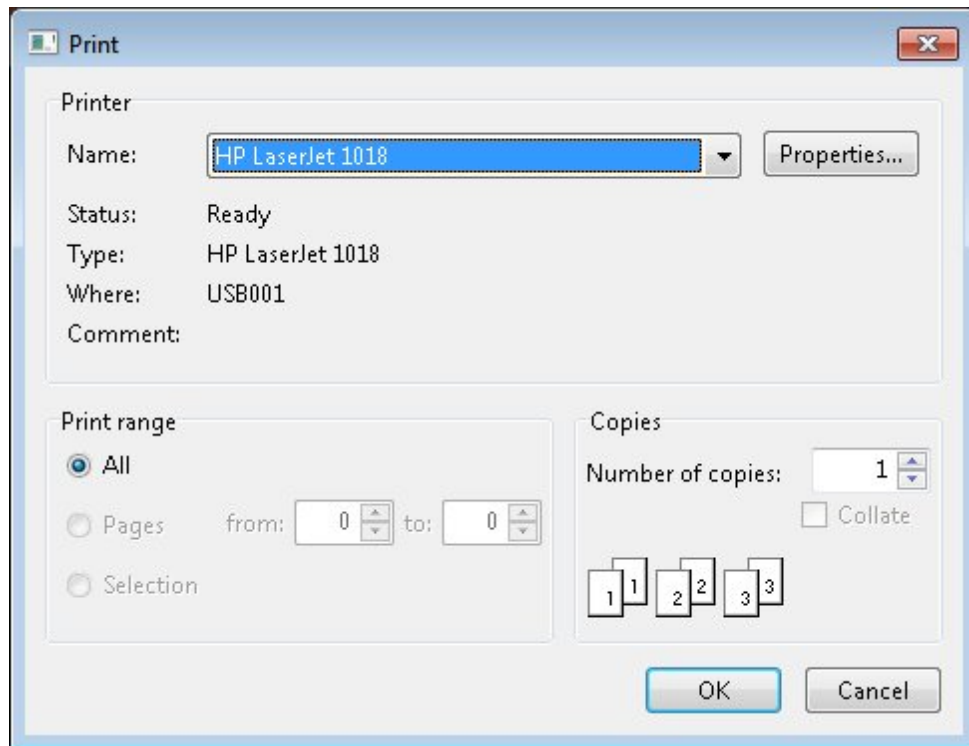
14.12 Customizing the PRNSETUP Dialog

Did you know that you can customize the PRNSETUP box, and use themes to match your Form, Application, #WHERE, CHOOSE, DIALOG and PAUSE windows?

Try the following examples which demonstrate the use of PRNSETUP command to match your Form, Application, #WHERE, CHOOSE, DIALOG and PAUSE themes.

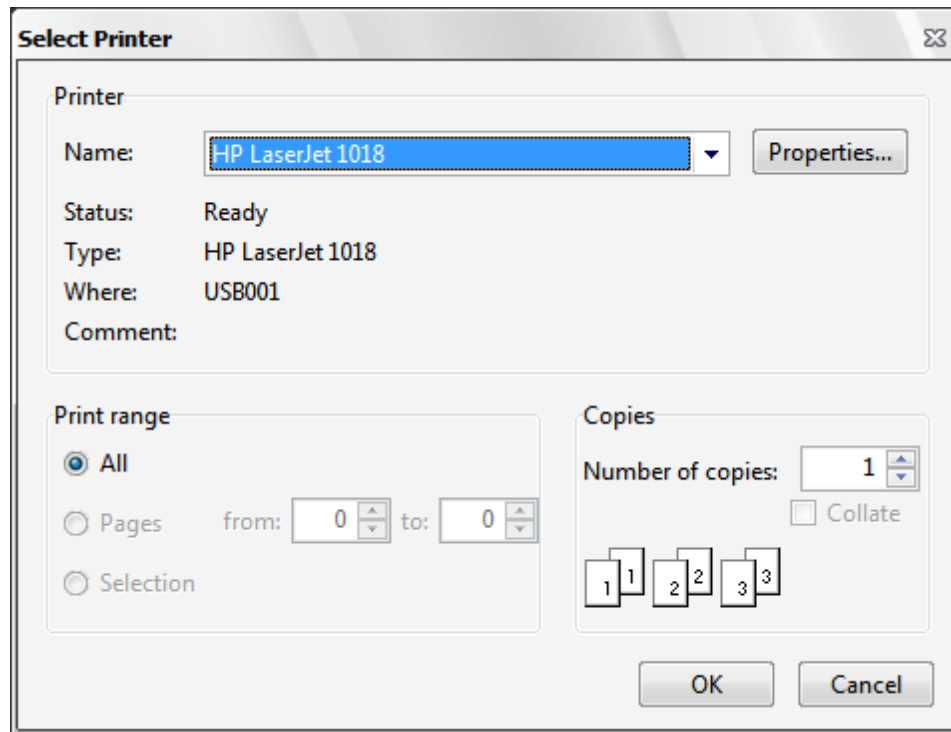
Example 01:

```
--(Changing Printer and then showing the window)
PRNSETUP 'Adobe PDF'
PRNSETUP
```



Example 02:

```
-- (Customized Caption with R:BASE Themes)
PRNSETUP OPTION CAPTION Select Printer|THEMENAME Vista CG
```

**Notes:**

01. If you use a printer name that does not exist there will be no error and the current printer will not be changed.
02. Use RBTI_PRNSETUP variable to capture user action. When using the PRNSETUP window, clicking on [OK] button will return the value of RBTI_PRNSETUP as "OK". When using the PRNSETUP window, clicking on [Cancel] button will return the value of RBTI_PRNSETUP as "CANCEL".
03. Use PRNSETUP INFO command at the R> prompt for detailed info on your currently selected printer.
04. For complete details, PRNSETUP parameters and OPTIONS: **HELP PRNSETUP**

14.13 Defining Font Name/Size with OUTPUT PRINTER

Did you know that you can define the font name and size when using the OUTPUT PRINTER command?

R:BASE allows you to define font name and font size when using the OUTPUT PRINTER command at the R> prompt, in a command file or in an EEP used in Forms, Reports and Labels.

Example 01

```
CONNECT RRBYW19
SET LINES 0
SET WIDTH 120
SET NULL ' '
OUTPUT PRINTER FONTNAME COURIER FontSize 9
SELECT +
Company=38 AS `Company Name`, +
CustAddress=32 AS `Address`, +
(CustCity+', '&CustState&CustZip)=30 AS `City, State and Zip` +
FROM Customer ORDER BY Company
OUTPUT SCREEN
SET LINES 20
SET WIDTH 79
```

Example 02

```
CONNECT RRBYW19
SET LINES 0
SET WIDTH 120
SET NULL ' '
OUTPUT PRINTER FONTNAME COURIER FontSize 8
SELECT +
Company=38 AS `Company Name`, +
CustAddress=32 AS `Address`, +
(CustCity+', '&CustState&CustZip)=30 AS `City, State and Zip`, +
CustPhone=14 AS `Phone Number` +
FROM Customer ORDER BY Company
OUTPUT SCREEN
SET LINES 20
SET WIDTH 70
```

14.14 Differentiating Between Regular and Temporary Tables/Views

R:BASE includes the new feature of differentiating between Regular and Temporary Tables/Views when using the LIST or LIST TABLES command at the R> prompt.

To safely indicate which tables are temporary tables in the LIST or LIST TABLES command, you will now see a "(T)" in front of the table name.

Example:

1. Launch R:BASE
2. CONNECT Concomp
3. Switch to R> Prompt and create a TEMPorary table as following:

```
PROJECT TEMP tCustomer FROM Customer USING ALL
LIST TABLES
```

You will notice the **(T)** in front of the tCustomer table.

4. Now create a TEMPorary VIEW as following:

```
CREATE TEMP VIEW `tYTDInvoiceTotal` +
(CustID, YTDInvoiceTotal) AS +
SELECT CustID, (SUM(InvoiceTotal)) FROM TransMaster +
GROUP BY CustID
COMMENT ON VIEW `tYTDInvoiceTotal` IS +
'Year-To-Date Invoice Total by Customer'
LIST
```

You will notice **(T)** in front of the tYTDInvoiceTotal view.

Note: All TEMPorary Tables/View created during individual R:BASE session disappear when the database is DISConnected.

14.15 Displaying Longer Table/Column Names

If you have a table or column name that is more than the traditional 18 characters wide, and if you wish to display a list of TABLES or COLUMNS, use the NAMEWIDTH setting.

The default setting for this exclusive NAMEWIDTH option is set to 18. However, you can change the name width to display table or columns names that are more than 18 characters to list them accordingly.

Syntax:

```
SET NAMEWIDTH nnn
```

Example:

```
SET WIDTH 120
SET NAMEWIDTH 40
SET LINES 0
LIST COLUMNS
```

14.16 Dropping Multiple Views

Did you know that you can drop multiple views with a single DROP command?

Here's how:

The DROP command has been enhanced to drop multiple views with the same DROP command by supporting the delimiter character between view names in the syntax. The R:BASE default delimiter is a comma (,).

```
DROP VIEW ViewName1, ViewName2, ViewName3
```

The NOCHECK parameter can also be added to eliminate the confirmation message.

```
DROP VIEW ViewName4, ViewName5, ViewName6 NOCHECK
```

14.17 Encrypting an UNLOADED DATA File

Traditionally, the UNLOAD command is used to unload the structure, data, structure with data, forms, labels or reports to a specified OUTPUT file. Depending on the option used, the UNLOAD command also creates a LOB file with matching output file name.

Typically, you can view the entire ASCII unloaded data using text editor. However, it becomes critical when the unloaded file contains sensitive data, including owner and user passwords, etc.

Did you know that you can encrypt the entire file using the new built-in 512-bit ENCRYPT option?

Example 01:

```
CONNECT RRBYW19
OUTPUT RRBYW19.str ENCRYPT mypassword
UNLOAD STRUCTURE
OUTPUT SCREEN
```

This will create an encrypted password-protected structure file (RRBYW19.str).

Example 02:

```
CONNECT RRBYW19
```



```
OUTPUT RRBYW19.all ENCRYPT mypassword
UNLOAD ALL
OUTPUT SCREEN
```

This will create an encrypted password-protected data RRBYW19.all and RRBYW19.lob file.

Example 03:

```
CONNECT RRBYW19
OUTPUT employee.asc ENCRYPT mypassword
UNLOAD DATA FOR Employee USING +
EmpID,EmpFName,EmpLName,EmpAddress,EmpAddress2,EmpCity,EmpState, +
EmpZip,EmpPhone,HireDate,EntryDate,EMailAddress AS ASCII
OUTPUT SCREEN
```

This will create an encrypted password-protected ascii employee.asc file.

While using the built-in REdit or external R:BASE Editor, you may also create/edit any external command file/EEP and add the encryption by using the "Add Encryption" when saving or using the "Save As" option.

An output file created with "ENCRYPT yourpassword" option can only be opened using R:BASE for Windows.

Attempting to edit or run the encrypted file will prompt you to enter the appropriate password.

14.18 Enhanced DIALOG Command

Did you know that you can include end-user "hints", apply a "theme", and use "lookup" options for pop-up menus when using the DIALOG command? These are just a few new enhancements in the DIALOG command.

When you double-click on the DIALOG edit box, not only can you bring up the customized lookup pop-up menus, but when you hover your mouse cursor over the dialog edit box, the hint can be displayed.

Here's how:

1. DIALOG_EDIT_HINT value, specifies the hint.
2. POPUP_ENABLED (ON/OFF), enables the option of bringing up a custom pop-up menu when using the double-click.
3. POPUP_DIALOG_TYPE value, specifies the Pop-up type when double-clicking on the DIALOG edit box.

Example: (Dialog with Hint and Pop-ups as Lookup)

```
-- Dialog_With_Hint_And_Popups_As_Lookup.RMD
-- Popup Dialog Type: LOOKUP
-- Along with DIALOG_EDIT_HINT value ...
CONNECT RRBYW19
CLEAR VAR vCaption,vResponse,vEndKey,vCustID,vRows, +
ivRows,vPauseMessage
SET VAR vCaption TEXT = 'DIALOG with Hint and Pop-up Menus'
SET VAR vResponse TEXT = NULL
SET VAR vEndkey TEXT = NULL
SET VAR vCustID INTEGER = NULL
SET VAR vRows INTEGER = NULL
SET VAR vPauseMessage TEXT = NULL
LABEL GetCustID
```

```

CLS
DIALOG 'Enter Customer ID' vResponse=33 vEndKey 1 +
  CAPTION .vCaption +
  ICON APP +
  OPTION DIALOG_EDIT_HINT Double-Click for List of Customers +
  | POPUP_ENABLED TRUE +
  | POPUP_DIALOG_TYPE LOOKUP +
  | POPUP_TITLE_FONT_NAME VERDANA +
  | POPUP_TITLE_FONT_SIZE 9 +
  | POPUP_TITLE_FONT_BOLD TRUE +
  | POPUP_TITLE_FONT_COLOR WHITE +
  | POPUP_TITLE_BACK_COLOR 5142690 +
  | POPUP_TITLE_TEXT Select Customer +
  | POPUP_LIST_FONT_COLOR WHITE +
  | POPUP_LIST_BACK_COLOR 5142690 +
  | POPUP_CAPTION List of Customers +
  | POPUP_TABLE Customer +
  | POPUP_SELECT Company,CustCity+', '&CustState&CustZip +
  | POPUP_RETURN_COLUMN CustID +
  | POPUP_WHERE ORDER BY Company +
  | POPUP_SHOW_LINES TRUE +
  | POPUP_RETURN_TYPE COLUMN +
  | POPUP_LINES 20 +
  | POPUP_DISTINCT TRUE +
  | THEMENAME R:BASE Rocks!
IF vEndKey = '[Esc]' THEN
  GOTO Done
ENDIF
IF vResponse IS NULL THEN
  PAUSE 2 USING 'Missing Customer ID!' +
  CAPTION .vCaption ICON STOP +
  BUTTON 'Click here to enter Customer ID again ...' +
  OPTION BACK_COLOR WHITE +
  | MESSAGE_COLOR WHITE +
  | MESSAGE_FONT_NAME Verdana +
  | MESSAGE_FONT_COLOR RED +
  | MESSAGE_FONT_SIZE 9 +
  | MESSAGE_FONT_BOLD ON +
  | MESSAGE_FONT_ITALIC OFF +
  | MESSAGE_FONT_STRIKEOUT OFF +
  | BUTTON_COLOR WHITE +
  | BUTTON_FONT_COLOR GREEN
  GOTO GetCustID
ENDIF
SET VAR vCustID = (INT(.vResponse))
CLS
SELECT COUNT(*) INTO vRows INDIC ivRows FROM +
  Customer WHERE CustID = .vCustID
IF vRows > 0 THEN
  SET VAR vFormMode = 'EDIT'
EDIT USING Customers WHERE CustID = .vCustID
ENDIF
SET VAR vResponse = NULL

```

```

SET VAR vCustID = NULL
GOTO GetCustID
LABEL Done
CLEAR VAR vCaption, vResponse, vEndKey, vCustID, vRows, +
    ivRows, vPauseMessage
RETURN

```

Customize or modify above code to meet your needs.

For more details, options and examples, please refer to:

From The Edge: <http://www.razzak.com/fte>
 Topic: All About DIALOG Command

14.19 Enhanced LIST Command Parameters

Did you know that now you can selectively LIST only LOCKed, dBASE, SERVER, and TEMP tables?

R:BASE supports several LIST command parameters:

```

-- Example 01 (to list only locked tables)
LIST TABLE LOCK

-- Example 02 (to list only dBASE tables)
LIST TABLE dBASE

-- Example 03 (to list only ODBC SATTACHed Server tables)
LIST TABLE SERVER

-- Example 04 (to list only temporary tables)
LIST TABLE TEMP

```

14.20 Enhanced LIST FORMS, REPORTS and LABELS Command

When a Form, Label, Report, or External Form is modified using the appropriate designer, having the ability to track and visually display the Date/Time Stamp and Version Number is among the many eXclusive features of R:BASE eXtreme 9.5 (64) Database Explorer.

Did you know that using the LIST FORMS, LABELS, or REPORTS command at the R> prompt now includes "Modified" and "Version" columns in addition to Name, Table/View, and Description?

```

-- Example 01 (To list all forms)
LIST FORMS

-- Example 02 (To list all forms for a given table)
LIST FORMS FOR tablename

-- Example 03 (To list all labels)
LIST LABELS

-- Example 04 (To list all labels for a given table)
LIST LABELS FOR tablename

-- Example 05 (To list all reports)
LIST REPORTS

-- Example 06 (To list all reports for a given table)
LIST REPORTS FOR tablename

```

14.21 JOIN, INTERSECT, SUBTRACT, and UNION - Temporary Table Support

Did you know that a TEMPORARY option has been enhanced for the JOIN, INTERSECT, SUBTRACT, and UNION commands, to support the creation of a temporary table?

The following example uses the JOIN command to create ProdDetail as a temporary table:

```
JOIN ProdLocation USING Model WITH Product USING Model +  
FORMING TEMPORARY ProdDetail
```

The following example uses the INTERSECT command to create RepSales as a temporary table:

```
INTERSECT Employee WITH SalesBonus FORMING TEMPORARY RepSales +  
USING EmpLname, EmpID, HireDate, NetAmount, TransDate, Bonus, BonusPct
```

The following example uses the SUBTRACT command to create NoBonus as a temporary table:

```
SUBTRACT SalesBonus FROM Employee FORMING TEMPORARY NoBonus
```

The following example uses the UNION command to create SalesRepHistory as a temporary table:

```
UNION Employee WITH SalesBonus FORMING TEMPORARY SalesRepHistory
```

14.22 LAUNCHing Operations with Show Modes

Did you know that you can open and utilize primary operations of the Windows operating system with the LAUNCH command, and control the "show mode" for everything being launched?

The LAUNCH command supports several new parameters for operating system operations and show modes for the launched program. New operation parameters allow for parts of the operating system to be utilized within R:BASE.

- CMD_EXPLORE - displays the Windows Explorer from the specified folder
- CMD_FIND - initiates a search from the specified folder
- CMD_PRINT - prints a file to the default printer, or displays a print dialog, based upon file type

Examples:

```
LAUNCH CMD_EXPLORE | C:\RBTI
```

```
LAUNCH CMD_FIND | F:\DATA
```

```
LAUNCH CMD_PRINT | C:\ReadMe.txt
```

Several show modes are available for the launched program or operation.

- NORMAL - launches the program in a normal window
- MINIMIZED - launches the program in a minimized window
- MAXIMIZED - launches the program in a maximized window
- HIDE - launches the program, but it is hidden from display

Examples:

```
LAUNCH C:\MyDocs\ReadMe.txt NORMAL
```

```
LAUNCH 'winword' | "C:\My Documents\AllFiles.docx" "/mMacroName" | W' MAXIMIZED
```

```
LAUNCH D:\Customers\Listing.doc MINIMIZED
```

```
LAUNCH CMD_PRINT | N:\Sales\OrderForm.pdf HIDE
```

14.23 Listing Database Keys

Did you know that you can use the new LIST command to list PRIMARY, FOREIGN and UNIQUE Keys?

Here's how:

1. LIST PKEYS

Using LIST PKEYS will return a list of all PRIMARY KEYS and PRIMARY KEYS REFERENCED.

2. LIST PKEYS FOR tablename

Using LIST PKEYS FOR tablename will return the name of a PRIMARY KEY column for a given table name and associated REFERENCED keys

3. LIST FKEYS

Using LIST FKEYS will return a list of all FOREIGN KEYS in the connected database.

4. LIST FKEYS FOR tablename

Using LIST FKEYS FOR tablename will return a list of all FOREIGN KEY columns for a given table.

5. LIST UKEYS

Using LIST UKEYS will return a list of all UNIQUE KEYS in the connected database.

6. LIST UKEYS FOR tablename

Using LIST UKEYS FOR tablename will return a list of all UNIQUE KEY columns for a given table.

14.24 Listing Tables with Cascade and Cascade Type

Did you know that you can list all tables with Cascade and Cascade Types, if defined, using the LIST command?

```
LIST CASCADE
```

LIST CASCADE will display a list of all tables with a CASCADE and type of cascade (UPDATE, DELETE or BOTH).

14.25 Managing Monitors and Monitor Properties

Did you know that there are new GETPROPERTY command parameters to help manage multi-monitor hardware environments?

For instance, the functionality to return the number of monitors connected to a system, is available using the GETPROPERTY command as follows:

```
GETPROPERTY APPLICATION MONITOR_COUNT vMonitorCount
```

In addition, the following GETPROPERTY command has been implemented to return the monitor screen size properties, using the value of X as a zero based index.

```
GETPROPERTY APPLICATION MONITOR_PROPERTY[X] VarName
```

The result is a comma separated list of values in the following format:

```
MonitorNumber,MonitorIsPrimaryYN,MonitorLeft,MonitorTop,MonitorWidth,MonitorHeight,MonitorWorkareaLeft,MonitorWorkareaTop,MonitorWorkareaRight,MonitorWorkareaBottom
```

The "WorkArea" properties is the monitor area, less the task bar area.

Example:

```
GETPROPERTY APPLICATION MONITOR_PROPERTY[0] vMonitorProps
SHOW VAR vMonitorProps
0,Y,0,0,1920,1080,0,0,1920,1040
```

Supported Versions:

- R:BASE X (Build..: 10.0.2.11130 or higher)

14.26 Programmatically Clean Scratch Files on Exit

R:BASE allows you to "Clean Scratch Files on Exit" by using an option within the R:BASE Settings; select "Settings" > "Clean Scratch Files on Exit" from the main Menu Bar.

However, when using R:Compiler compiled executables or Runtime applications, this option is not available, obviously for restricted access to all settings, designers, and the R> Prompt.

Did you know that through your code you can clean the scratch files on exit, using the following PROPERTY command in the application startup file, as an "On Before Start" Custom EEP in the main application form, External Form File, or R:BASE Application (.RBA) file?

```
PROPERTY APPLICATION CLEANSCRATCHFILESONEXIT 'TRUE'
RETURN
```

Using this option will automatically clean all scratch files upon exiting of your compiled or runtime application.

14.27 RECALCulating Specific Variables

Traditionally, using the RECALC VARIABLES command recalculates all variables.

Did you know that now you can recalculate a specific variable or a list of variables?

The RECALC VARIABLES command has been enhanced to include an optional variable list, to perform a recalculation on a target variable or set of specified variables.

```
-- Example 01:
RECALC VARIABLES vTargetPrice
RETURN

-- Example 02:
RECALC VARIABLES vTargetPrice,vShippedQuantity,vDiscountedRate
RETURN
```

Supported Versions:

- R:BASE X (Build..: 10.0.1.10727 or higher)

14.28 RUN filespec vs. QUIT TO filespec

Creating complex applications with many command files and monitoring nesting levels has always been an art with designing trouble-free applications. It is the programmer's responsibility to monitor the nesting level, when using RUN filespec command from within a command or procedure file.

When you use the RUN filespec command from within a command or procedure file, R:BASE adds one to the nesting level for each nested command file. The RETURN command is the only way to exit and to return control to the next line in the calling command file and to decrease the nesting counter by one.

You can nest command files, such as RUN filespec, IF structures, WHILE loops, and SWITCH structures. The number of control structures that you can nest depends on the amount of memory available when the command or procedure file executes.

You can have up to five levels of RUN execution, unless you use the SET FILES command to change the number of command files that can be open simultaneously. In other words, command file 1 can execute command file 2, command file 2 can execute command file 3, command file 3 can execute command file 4, and command file 4 can execute command file 5. Successive RETURN commands pass execution control back up through the command files.

It all comes down to defining nesting levels and ... making sure to return control back to the originating command file.

Using the QUIT TO filespec command, there is no overhead, no nesting level, nor does R:BASE have to keep track of nesting levels.

Using the QUIT TO filespec command, you can QUIT TO any command file at any level you wish and never have to worry about returning to the last command to return control to the one command before and so on ...

QUIT TO filespec command clears all nesting levels and resets the nesting level to 0. Thus, no overhead.

You can use QUIT TO in individual binary codelocked command file(s), however, QUIT TO filespec cannot be used to execute a command block in a procedure file.

14.29 Saving the Clipboard Contents to an External File

Did you know you can save the Windows Clipboard contents to an external file?

A new PROPERTY command CLIPBOARD object has been introduced with support to save Windows Clipboard contents to a text file, a binary file, and also the capability to clear the contents of the clipboard.

Please review the following syntax:

```
PROPERTY CLIPBOARD SAVE_TO_TEXT_FILE 'FileName.txt'  
PROPERTY CLIPBOARD SAVE_TO_BINARY_FILE 'FileName.bmp'  
PROPERTY CLIPBOARD CLEAR 'TRUE'
```

Use SAVE_TO_TEXT_FILE for saving the clipboard contents to a text file.

Use SAVE_TO_BINARY_FILE for binary data, like when pressing [Print Screen].

Use CLEAR to clear the clipboard contents.

Note:

Not all clipboard contents can be saved as binary data because the clipboard does not dictate a specific format to be used.

Supported Versions:

- R:BASE X (Build...: 10.0.2.11130 or higher)

14.30 Selecting Available Printers for R:BASE Sessions

When printer names change and new printers are installed, here's a code sample that you can use to select the appropriate printer for the current R:BASE session, regardless of long printer names, printer names with spaces, or printer with a UNC path.

```
-- Start here  
-- Selecting Available Printer for R:BASE Sessions
```

```

-- Author: A. Razzak Memon
CLEAR VAR vAvailablePrinters,vItemCount,vSelectedPrinter
SET VAR vAvailablePrinters NOTE = NULL
SET VAR vItemCount INTEGER = NULL
SET VAR vAvailablePrinters = (CVAL('Printers'))
SET VAR vItemCount = (ITEMCNT(.vAvailablePrinters))
SET VAR vSelectedPrinter TEXT = NULL
CLS
CHOOSE vSelectedPrinter FROM #LIST .vAvailablePrinters CHKBOX 1 +
TITLE 'Select Appropriate Printer and Click on [OK] Button to Continue' +
CAPTION 'Available Printers' +
LINES .vItemCount FORMATTED +
OPTION WINDOW_BACK_COLOR WHITE +
|TITLE_BACK_COLOR WHITE +
|TITLE_FONT_NAME Tahoma +
|TITLE_FONT_COLOR NAVY +
|TITLE_FONT_SIZE 14 +
|LIST_BACK_COLOR WHITE +
|LIST_FONT_NAME Tahoma +
|LIST_FONT_COLOR NAVY +
|LIST_FONT_SIZE 12 +
|BUTTONS_SHOW_Glyph ON +
|BUTTONS_BACK_COLOR WHITE
IF vSelectedPrinter IS NULL OR vSelectedPrinter = '[Esc]' THEN
    GOTO Done
ENDIF
PRNSETUP .vSelectedPrinter
LABEL Done
CLEAR VAR vAvailablePrinters,vItemCount,vSelectedPrinter
RETURN
-- End here

```

For a complete list of PRNSETUP OPTIONS, refer to HELP PRNSETUP.

14.31 Selecting the Appropriate Tray Using PRNSETUP

Did you know that you can select the printer specific paper tray using the PRNSETUP command?

Syntax:

```
PRNSETUP TRAY_NAME <Printer Specific Tray name>
```

-- Example 01: (to select printer specific Tray1)

```
PRNSETUP TRAY_NAME Tray1
RETURN
```

-- Example 02: (to select printer specific Tray2)

```
PRNSETUP TRAY_NAME Tray2
RETURN
```

-- Example 03: (to select printer specific MP Tray)

```
PRNSETUP TRAY_NAME MP Tray
```


RETURN

14.32 Turning the Data Browser into a Quick Form

Did you know that you can turn the Data Browser into a quick form?

You can achieve that by using the "ROWVIEW" option of the BROWSE command.

Here's how:

1. Start R:BASE
2. CONNECT ConComp
3. Create a command file with the following syntax:

```
-- DataBrowserAsForm.RMD
CLS
BROWSE * FROM Customer +
  ORDER BY Company +
  NOCHG +
  OPTION WINDOW_STATE MAXIMIZED +
  |CAPTION ConComp Customers +
  |ROWVIEW ON
RETURN
```

4. RUN DataBrowserAsForm.RMD

For enhanced database security from users with the same command, you can use disable any output options.

3. Create a command file with the following syntax:

```
-- DataBrowsSecure.RMD
CLS
BROWSE * FROM Customer +
  ORDER BY Company +
  NOCHG +
  OPTION WINDOW_STATE MAXIMIZED +
  |CAPTION ConComp Customers +
  |ROWVIEW ON +
  |EXPORTRESULT OFF +
  |SAVERESULT OFF +
  |PRINTDATA OFF
RETURN
```

4. RUN DataBrowsSecure.RMD

14.33 Understanding Dotted vs. Ampersand Variables

Dotted Variables

1. Variables used to hold values.
2. When you want R:BASE to use the "value" contained in the variable.
3. When you use a variable in a calculation or as a comparison value, you use "dot" variable.
4. A rule of thumb for when to "dot" a variable is to always "dot" the variable when it is on the right side of the operator.
5. Dotted a variable basically turns it into a constant value. R:BASE looks only at the value of the variable when it is dotted. R:BASE doesn't look at the datatype, just at the value the variable

contains. That's why you can have a TEXT datatype variable (result of a CHOOSE or DIALOG command, for example) containing a value that looks like a DATE and use that variable to compare to a DATE datatype column or variable.

6. In expressions, R:BASE checks the datatype of a dotted variable. An expression is anything enclosed in parentheses. R:BASE verifies the datatype in expressions to make sure the expression is valid. You can't add an INTEGER to a TEXT for example.
7. Dotted variables are used in Form and Report expressions as they are in the SET VAR command. On the right side of the operator, dot the variable.
8. Dotted variables are commonly used in WHERE clauses and in calculations with other variables.

Example 01:

```
SET VAR vDate DATE = 11/19/2016
SELECT * FROM TransMaster WHERE TransDate <= .vDate
```

The SELECT queries all of the records from the TransMaster table where the value in the TransDate column is less than or equal to the value contained in the variable vdate.

Example 02:

```
SET VAR vOrder INTEGER = 20001
SET VAR vAmount CURRENCY = NULL
SET VAR vShip TEXT = NULL
SET VAR vFreight CURRENCY = NULL
SET VAR vTax CURRENCY = NULL
SET VAR vState TEXT = NULL
SELECT NetAmount, ShipMethod, StateAbr INTO +
  vAmount INDIC IvAmount, +
  vShip INDIC IvShip, +
  vState INDIC IvState +
  FROM Orders WHERE OrderNum = .vOrder
IF vShip = 'AIR' THEN
  SET VAR vFreight = $11.00
ELSE
  SET VAR vFreight = $5.00
ENDIF
SET VAR vAmount = (.vAmount + .vFreight)
SELECT TaxRate INTO vTax INDIC IvTax +
  FROM States WHERE StateAbr = .vState
SET VAR vAmount = (.vAmount+(.vAmount*.vTax))
```

On the right side of the equals sign (the operator), the variable is dotted. On the left side of the operator, in the IF and the SET VAR commands, the variable is referenced by its name only, it is not dotted.

Amperand Variables

1. You can't dot a variable when it contains part of a command - a table or column name, or an ORDER BY or WHERE clause.
2. When a variable contains part of a command, its name is prefaced with an ampersand, "&", and it is called an ampersand variable. The "&" in front of the variable name tells R:BASE that the variable contains part of the command, not a value, and the contents of the variable are used when parsing the command.
3. Don't confuse the ampersand that prefaced a variable name with the ampersand that is used to concatenate TEXT values.
4. Because an ampersand variable is part of a command, it can't be used inside parentheses. Parentheses indicate expressions, expressions are parsed separately from the rest of the command. You need to include the parentheses as part of the variable value. Sub-selects and IN lists are enclosed in parentheses and you can't use an ampersand variable inside them, you need to include the entire sub-select or IN list, including parentheses, as the variable value.

5. Ampersand variables are most often used to hold table and column names and WHERE and ORDER BY clauses. By using ampersand variables to hold column and table names, you can use the same command to select data from different tables. The CHOOSE command displays menus of available tables and columns.

Example 01:

```
SET VAR vTable TEXT = NULL
SET VAR vColList TEXT = NULL
CLS
CHOOSE vTable FROM #TABLES AT CENTER,CENTER TITLE 'Choose Table' +
  CAPTION 'List of Tables' LINES 18 FORMATTED
CLS
CHOOSE vColList FROM #COLUMNS IN &vTable AT CENTER,CENTER +
  CHKBOX TITLE 'Choose Column(s)' CAPTION 'List of Columns' +
  LINES 18 FORMATTED
```

The values selected from the menus are placed into variables.

The variables might look like this:

```
vTable = Employee TEXT
vColList = Empid,EmpLname,EmpPhone,EmpExt TEXT
```

The variables are then used in any command that uses a table name or column list. Where you see Column names or Table/View name you can substitute an ampersand variable that contains the Column, Table or View name. The variables must be used as ampersand variables to tell R:BASE they contain part of the command.

Example 02:

```
BROWSE &vColList FROM &vTable
```

To prompt for an ORDER BY clause, use the CHKSORT option instead of CHKBOX on the CHOOSE...FROM #COLUMNS.

Example 03:

```
SET VAR vOrderBy TEXT = NULL
CHOOSE vOrderBy FROM #COLUMNS IN &vTable AT CENTER,CENTER +
  CHKSORT TITLE 'Choose Column(s)' CAPTION 'Order By' +
  LINES 18 FORMATTED
```

The CHKSORT option prompts for Ascending or Descending just like the R:BASE sort menus. The variable contains ASC or DESC as well as the column names. It might look like this:

```
vOrderBy = EmpLname ASC,EmpFname ASC TEXT
```

In the command, follow the keywords ORDER BY with the ampersand variable containing the columns to order by.

Example 04:

```
BROWSE &vColList FROM &vTable ORDER BY &vOrderBy
```

14.34 Understanding WHILE Loop Optimization

Have you ever run into a situation where your command file/application is not working correctly by variables not being calculated accurately or by experiencing unexpected closes within a WHILE..ENDWHILE loop; and when you TRACE or RUN that routine with WHILEOPT set to off, R:BASE returns no errors and everything works fine?

Well, those are the typical symptoms of WHILE loop optimization. There is likely a problem with an optimized command and the problem is most likely with a variable that you may be clearing, redefining, or not data typing.

WHILE loops have always been considered one of the faster ways to process code as R:BASE places the WHILE loop code into memory and then parses it before beginning execution. R:BASE does not need to perform line-by-line reading and parsing of the code for each iteration of the WHILE loop. As the code is read into memory, the WHILE loop code was parsed into tokens (4-byte segments). But, R:BASE still needed to figure out what was what -- parse the expressions, find variable names and values, etc. It was faster than reading and parsing line-by-line, but not as fast as it could be.

If you determine that the problem is with an optimized command, review the rules below:

1. Pre-define all variable(s) with NULL values at the beginning of a command file or before the WHILE loop is executed.
2. Always use DROP CURSOR cursor before DECLARE CURSOR.
3. When using FETCH cursor INTO varname, always use INDICATOR variables for each defined variable(s).
4. Use WHILE SQLCODE <> 100 THEN
5. Never CLEAR any variable(s) within the WHILE loop. The value can be reset to NULL, but not cleared.
6. Never change variable data types within the WHILE loop.
7. Make sure to DROP CURSOR cursor after the WHILE loop.
8. Make sure to CLEAR unwanted and INDICATOR variables after the WHILE loop.

Keep in mind that TRACE does not use the WHILE loop optimization.

Example:

```
SET VAR vVarName ColType = NULL
DROP CURSOR C#1
DECLARE C#1 CURSOR FOR SELECT ColName FROM TableName +
    ORDER BY ColName
OPEN C#1
FETCH C#1 INTO vVarName INDIC IvVarName
WHILE SQLCODE <> 100 THEN
    -- Do what you have to do here with that sorted record
    FETCH C#1 INTO vVarName INDIC IvVarName
ENDWHILE
DROP CURSOR C#1
CLEAR VAR vVarName, IvVarName
```

14.35 Using Additional Data Browser/Editor Options

When using the BROWSE/EDIT commands, the following OPTION parameters provide the ability to customize the Data Browser/Editor Window:

- WINDOW_STATE value (MAXIMIZED, NORMAL, MINIMIZED)
- CAPTION value
- LOCK nnn
- ROWVIEW
- READONLY value (ON, OFF)
- MODAL value (ON, OFF)

- EXPORTRESULT value (ON, OFF)
- SAVERESULT value (ON, OFF)
- PRINTDATA value (ON, OFF)

When "BROWSE/EDIT columnlist FROM tableviewname" command includes BLOB (VARCHAR, BITNOTE, VARBIT) data, typically you will notice a white square box (indication of NOT NULL field) or white square box with red "X" (indication of NULL or empty field). With the full development version of R:BASE installed, you may change the "Layout" settings for Data Browser/Editor to "Show NOTE Hints" and "Show Image Hints" when mousing over the column cells with BLOB data.

What if you are using a compiled/runtime version of your R:BASE application and would like to provide these additional options?

Did you know that you can dynamically enable or disable these additional options using the following parameters in your R:BASE applications?

- ZEBRA_STRIPE value (ON, OFF) - specifies if zebra stripes are displayed
- CELL_HINTS value (ON, OFF) - specifies if NOTE column cell hints are displayed
- IMAGE_HINT value (ON, OFF) - specifies if image hints are displayed

```
-- Example 01 (Display Zebra Stripes)
CONNECT RRBYW19
BROWSE * FROM Contact OPTION ZEBRA_STRIPE ON

-- Example 02 (Display Cell Hints)
CONNECT RRBYW19
BROWSE * FROM Contact OPTION CELL_HINTS ON

-- Example 03 (Display Image Hints)
CONNECT RRBYW19
BROWSE * FROM Contact OPTION IMAGE_HINT ON

-- Example 04 (Display Zebra Stripes, Cell Hints, Image Hints)
CONNECT RRBYW19
BROWSE * FROM Contact +
OPTION ZEBRA_STRIPE ON|CELL_HINTS ON|IMAGE_HINT ON
```

14.36 Using an Enhanced Calculator in a Dialog Box

Did you know that you can bring up the enhanced R:BASE Calculator by simply double-clicking in a DIALOG box?

This newly introduced enhancement also adapts the R:BASE Themes, if used as an option for the DIALOG command.

```
-- Example 01:
-- Dialog_With_Popup_As_Calculator_01.RMD
CLS
CLEAR VAR vValueTxt,vValue,vEndKey
SET VAR vValueTxt TEXT = NULL
SET VAR vValue DOUBLE = NULL
DIALOG 'Enter Value or Double-Click for Calculator' +
vValueTxt=26 vEndKey 1 +
CAPTION 'DIALOG with Calculator Pop-up' ICON APP OPTION +
POPUP_ENABLED TRUE +
|POPUP_DIALOG_TYPE CALCULATOR
SET VAR vValue = .vValueTxt
```

```

CLEAR VARIABLE vValueTxt,vEndKey
RETURN

-- Example 02:
-- Dialog_With_Popup_As_Calculator_02.rmd
-- Using THEMENAME as Razzmatazz
-- Notice the dialog box and the matching calculator
CLS
CLEAR VAR vValueTxt,vValue,vEndKey
SET VAR vValueTxt TEXT = NULL
SET VAR vValue DOUBLE = NULL
DIALOG 'Enter Value or Double-Click for Calculator' +
  vValueTxt=36 vEndKey 1 +
  CAPTION 'DIALOG with Calculator Pop-up' ICON APP OPTION +
  POPUP_ENABLED TRUE +
  |POPUP_DIALOG_TYPE CALCULATOR +
  |THEMENAME Razzmatazz
SET VAR vValue = .vValueTxt
CLEAR VARIABLE vValueTxt,vEndKey
RETURN

-- Example 03:
-- Dialog_With_Popup_As_Calculator_03.rmd
-- Using THEMENAME as R:BASE Rocks!
-- Notice the dialog box and the matching calculator
CLS
CLEAR VAR vValueTxt,vValue,vEndKey
SET VAR vValueTxt TEXT = NULL
SET VAR vValue DOUBLE = NULL
DIALOG 'Enter Value or Double-Click for Calculator' +
  vValueTxt=30 vEndKey 1 +
  CAPTION 'DIALOG with Calculator Pop-up' ICON APP OPTION +
  POPUP_ENABLED TRUE +
  |POPUP_DIALOG_TYPE CALCULATOR +
  |THEMENAME R:BASE Rocks!
SET VAR vValue = .vValueTxt
CLEAR VARIABLE vValueTxt,vEndKey
RETURN

```

14.37 Using AS ALIAS and CAPTION With External Forms

Did you know that you can now specify an alias name when running an external form file as a modeless form, and include a dynamic caption as well?

The REFF command, to run an external form file, has been enhanced to support the AS ALIAS and CAPTION parameters.

Syntax:

```
REFF ExternalForm.rff MDI AS ExtForm_AliasName CAPTION 'Form Caption'
```

MDI - Specifies a modeless form, allowing users to access other windows without closing the form first.

AS alias - Allows the user to specify a name for this instance of the form. This option is used in conjunction with the MDI option so that the user can programmatically specify the form with the SETFOCUS command.

CAPTION 'windowcaption' - Specifies a caption for the window title bar.

Here's how:

```
REFF RRBYW19.RFF CAPTION 'Running R:BASE Your Way!'

REFF AdhocReports.RFF MDI AS Adhoc CAPTION 'Adhoc Custom Reports'
```

14.38 Using DROP ALL

Typically, the DROP ALL TABLES command can be used to DROP all tables with a warning dialog prompt (Yes/No) to verify each table and DROP ALL VIEWS command can be used to DROP all views with a warning dialog prompt (Yes/No) to verify each view.

However, if you wish to DROP ALL tables except a few, you may use the DROP ALL TABLES EXCEPT tablename1, tablename2 command.

You may also use the **enhanced** DROP ALL VIEWS EXCEPT viewname1, viewname2 command.

Summary:

- Use DROP ALL or DROP ALL TABLES command to drop all TABLES.
- Use DROP ALL VIEWS command to drop all VIEWS.
- Use DROP ALL TABLES EXCEPT listoftables command to drop all tables except the ones listed, separated by commas.

Example:

```
DROP ALL TABLES EXCEPT tablename1, tablename2
```

- Use DROP ALL VIEWS EXCEPT listofviews command to drop all views except the ones listed, separated by commas.

Example:

```
DROP ALL VIEWS EXCEPT viewname1, viewname2
```

14.39 Using INPUT Color for the DIALOG Command

Did you know that you can use "INPUT_BACKGROUND_COLOR value" to customize the input box background color in DIALOG command?



Here's how:

```
-- Using_Input_Color_For_Dialog.RMD
-- Author: A. Razzak Memon
SET VAR vInput TEXT = NULL
```

```

SET VAR vEndKey TEXT = NULL
SET VAR vCaption TEXT = 'Using Input Color for DIALOG Box'
LABEL GetInput
CLS
SET VAR vInput = NULL
DIALOG 'Enter Message Text' vInput=30 vEndKey 1 +
CAPTION .vCaption ICON INFO +
OPTION MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_COLOR NAVY +
|MESSAGE_FONT_SIZE 11 +
|INPUT_BACKGROUND_COLOR YELLOW +
|THEMENAME Longhorn
IF vEndKey = '[Esc]' THEN
GOTO Done
ENDIF
IF vInput IS NULL THEN
PAUSE 2 USING 'Missing Message Text' +
CAPTION .vCaption ICON WARNING +
BUTTON 'Press any key to enter message text' +
OPTION MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_COLOR RED +
|MESSAGE_FONT_SIZE 11 +
|INPUT_BACKGROUND_COLOR YELLOW +
|THEMENAME Longhorn
GOTO GetInput
ENDIF
-- Do what you have to do here ...
LABEL Done
CLEAR VARIABLES vInput,vEndKey,vCaption
RETURN

```

14.40 Using INITPOS and INITVAL OPTIONS in CHOOSE Commands

Did you know that you can pre-select initial position(s) for the items as well as pre-select initial value(s) for the list of items in CHOOSE command?

R:BASE includes two new command parameters, INITPOS and INITVAL, as part of the CHOOSE OPTIONS.

INITPOS specifies the pre-selected initial position(s) for the items in the list. The first position in the list is assigned zero, and increments for the number of values in the list (0 = 1, 1 = 2, 3 = 4, etc).

INITVAL specifies the pre-selected initial value(s) for the items in the list.

```

-- Example 01
-- Using_INITPOS_Option_In_CHOOSE_Command.RMD
-- Author: A. Razzak Memon
IF (CVAL('DATABASE')) <> 'RRBYW19' OR (CVAL('DATABASE')) IS NULL THEN
CONNECT RRBYW19 IDENTIFIED BY NONE
ENDIF
SET VAR vTitle TEXT = 'Choose Table(s)'
SET VAR vCaption TEXT = 'Using INITPOS Option in CHOOSE Command'
SET VAR vChoose TEXT = NULL
CLS
CHOOSE vChoose FROM #TABLES CHKBOX 2 +

```

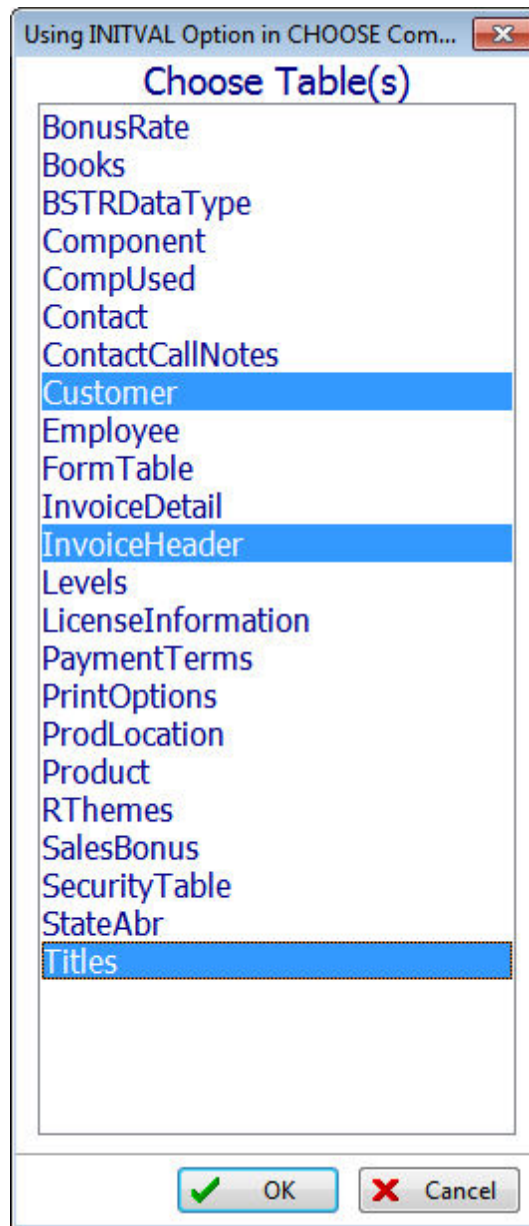


```
TITLE .vTitle +
CAPTION .vCaption +
LINES 27 +
OPTION INITPOS 2,5 +
|WINDOW_BACK_COLOR WHITE +
|TITLE_BACK_COLOR WHITE +
|TITLE_FONT_NAME Tahoma +
|TITLE_FONT_COLOR NAVY +
|TITLE_FONT_SIZE 14 +
|LIST_BACK_COLOR WHITE +
|LIST_FONT_NAME Tahoma +
|LIST_FONT_COLOR NAVY +
|LIST_FONT_SIZE 12 +
|BUTTONS_SHOW_Glyph ON +
|BUTTONS_BACK_COLOR WHITE
IF vChoose = '[Esc]' THEN
GOTO Done
ENDIF
-- Do what you have to do here ...
LABEL Done
CLEAR VARIABLES vTitle,vCaption,vChoose
RETURN
```



```
-- Example 02
-- Using_INITVAL_Option_In_CHOSE_Command.RMD
-- Author: A. Razzak Memon
IF (CVAL('DATABASE')) <> 'RRBYW19' OR (CVAL('DATABASE')) IS NULL THEN
CONNECT RRBYW19 IDENTIFIED BY NONE
ENDIF
SET VAR vTitle TEXT = 'Choose Table(s)'
SET VAR vCaption TEXT = 'Using INITVAL Option in CHOOSE Command'
SET VAR vChoose TEXT = NULL
CLS
CHOOSE vChoose FROM #TABLES CHKBOX 3 +
TITLE .vTitle +
CAPTION .vCaption +
```

```
LINES 27 +
OPTION INITVAL Customer,InvoiceHeader,Titles +
|WINDOW_BACK_COLOR WHITE +
|TITLE_BACK_COLOR WHITE +
|TITLE_FONT_NAME Tahoma +
|TITLE_FONT_COLOR NAVY +
|TITLE_FONT_SIZE 14 +
|LIST_BACK_COLOR WHITE +
|LIST_FONT_NAME Tahoma +
|LIST_FONT_COLOR NAVY +
|LIST_FONT_SIZE 12 +
|BUTTONS_SHOW_Glyph ON +
|BUTTONS_BACK_COLOR WHITE
IF vChoose = '[Esc]' THEN
GOTO Done
ENDIF
-- Do what you have to do here ...
LABEL Done
CLEAR VARIABLES vTitle,vCaption,vChoose
RETURN
```



14.41 Using ON CONNECT

Did you know that you can use the ON CONNECT command to save and reset your ON CONNECT parameters in R:BASE?

Syntax Sample 01:

```
ON CONNECT RUN commadfile SAVE
```

Where:

- CONNECT indicates the command file should be run when connecting a database.
- RUN specifies the name of the command file to run.
- commadfile specifies the name of the command file to run.
- SAVE indicates the defined ON CONNECT parameters be saved.

- Any options without the SAVE parameters will only be active for a session.

Example 01:

```
CONNECT RRBYW19 IDENTIFIED BY NONE
ON CONNECT RUN RRBYW19.DAT SAVE
```

To RESET the ON CONNECT parameters:

Syntax Sample 02.:

```
ON CONNECT RESET SAVE
```

Where:

- RESET turns the ON CONNECT procedure off.
- SAVE indicates the defined ON CONNECT parameters be saved.

Example 02:

```
CONNECT RRBYW19 IDENTIFIED BY NONE
ON CONNECT RESET SAVE
```

14.42 Using PRNSETUP Help Options

Did you know that you can use the PRNSETUP HELP options to find available Fonts, Printers, Formats, Sources and other details about your printer?

Here's how:

1. Start R:BASE
2. At the R> prompt, type:

```
PRNSETUP HELP
```

This will list all available options, such as:

```
FONTS - Available Fonts
PRINTERS - Available Printers
PAPERNAMEs - Available Formats
TRAYS - Available Sources
INFO - Current Printer Information
```

```
PRNSETUP FONTS
```

This will list all available fonts.

```
PRNSETUP PRINTERS
```

This will list all available printers. Printer name preceded by "*" denotes the default printer.

```
PRNSETUP PAPERNAMEs
```

This will list all available paper formats.

```
PRNSETUP TRAYS
```

This will list all available paper sources.

PRNSETUP INFO

This will list all details about the default printer, including:

- Current Printer
- Status
- Paper Orientation
- Paper Size
- Paper Source
- Print Quality (DPI)
- Copies
- Color Mode
- Duplex Mode
- Collation

14.43 Using the Default Tab for the R:BASE BLOB Editor

Did you know that R:BASE BLOB Editor RBBEDIT (RBBE) command has been enhanced to specify the default tab for the R:BASE BLOB Editor, if desired?

```
RBBEDIT [ filename.ext [ ZoomMode ][ DefaultTab ] ]
```

filename.ext

Specifies an external ASCII text, RTF, or image file. The R:BASE BLOB Editor recognizes the following image formats:

GIF	BMP
DCX	PCC
PCX	AFI
VST	TGA
PNG	JPEG
JPG	TIFF
TIF	ICO
EMF	WMF

DefaultTab

Specifies the default tab for the R:BASE BLOB Editor.

Default Tab	Parameter
Image	NOTE, VARCHAR, TEXT, NOTEVARCHAR
Note/Varchar	IMAGE
Rich Text	RICHTEXT, RTF
Unicode/BSTR	UNICODEBSTR, UNICODE, BSTR
Hex	HEX

Examples:

```
RBBEDIT test.txt TEXT
```

```
RBBEDIT image.bmp IMAGE
```

```
RBBEDIT filename HEX
```

```
RBBEDIT mydoc.rtf RICHTEXT
```

14.44 Using the Enhanced ATTACH Command

R:BASE's ATTACH command allows users to attach an external dBASE file to a connected R:BASE database.

Did you know that now you can totally automate the attachment of all your dBASE files using the enhanced ATTACH command?

In addition to using the traditional ATTACH command, and then manually answering all questions and ALIASing conflicting reserved column names, now you can enhance your routines by defining selective columns as well as ALIAS column names with all supported parameters just like SATTACH command, including the option to ATTACH dBASE file as TEMPORARY table.

```
ATTACH [ TEMPORARY ] filespec [ AS tablealias ] [ USING [ ndxlist ] ] [ ALIAS aliaslist ]
```

Options

ALIAS AliasList

To specify alias names for columns.

AS tablealias

Specifies an alias, or temporary name, for the dBASE table. A table alias is sometimes required when attaching files that do not follow the same table name restrictions as R:BASE.

filespec

A dBASE database name with a drive and path specification in the form D:\PATHNAME\FILENAME.

ndxlist

Specifies a list of index files to associate with the specified dBASE file. You do not have to include the extension for each index file. Separate index file names with a comma (or the current delimiter). Index files must be located with the specified dBASE file.

TEMPORARY

Allows you to create a temporary table with the ATTACH command. The temporary tables will disappear when the database is disconnected.

USING

Removes dBASE index files that were previously associated with the attached dBASE file when this option is used without a list of index files.

-- Example 01:

```
ATTACH TEMPORARY filename.dbf +
AS TableAlias +
ALIAS Column1, Column2, Column3, Column4, Column5
```

14.45 Using the Enhanced DIR Command in R:BASE

Did you know that the traditional DIR command has been enhanced to display new Windows output format? DIR command displays a list of a directory's files and sub-directories. Used without parameters, DIR displays the disk's volume label and serial number, followed by a list of directories and files on the disk, including their names and the date and time each was last modified.

For files, DIR displays the long name, extension and the size in bytes. DIR also displays the total number of files and directories listed, their cumulative size, and the free space (in bytes) remaining on the disk.

If you wish to display the traditional output format, you can use the following new SET SHORTNAME ON

command. The default is set to OFF.

In addition, new SHOW SHORTNAME command will display the current setting and new (CVAL('SHORTNAME')) function will retrieve the current setting when used in a command file.

The default R:BASE configuration files, when created from scratch, will also include the new default setting as SHORTNAM OFF.

14.46 Using the GAUGE Options with PAUSE

If you are running a routine and don't know exactly how long it is going to take, now you may use the new GAUGE options to display a cool gauge with custom theme and colors.

Here's how:

Example 01:

```
CLS
PAUSE 3 USING 'Calculating ... Please Stand By ...' +
CAPTION ' Pause 3 with Gauge ' ICON APP +
OPTION GAUGE_VISIBLE ON +
|GAUGE_COLOR [R218,G228,B246] +
|GAUGE_INTERVAL 10 +
|MESSAGE_FONT_NAME VERDANA +
|MESSAGE_FONT_SIZE 10 +
|MESSAGE_FONT_COLOR BLUE +
|THEMENAME Razzmatazz
-- Put your code, that takes a lot of time, here ...
-- Use CLS command to clear the PAUSE 3 with gauge
RETURN
```

Example 02:

```
CLS
PAUSE 3 USING 'Calculating ... Please Stand By ...' +
CAPTION ' Pause 3 with Gauge ' ICON APP +
OPTION GAUGE_VISIBLE ON +
|GAUGE_COLOR 2708091 +
|GAUGE_INTERVAL 10 +
|MESSAGE_FONT_NAME VERDANA +
|MESSAGE_FONT_SIZE 10 +
|MESSAGE_FONT_COLOR RED +
|THEMENAME R:BASE Rocks!
-- Put your code, that takes a lot of time, here ...
-- Use CLS command to clear the PAUSE 3 with gauge
RETURN
```

Example 03:

```
CLS
PAUSE 3 USING 'Calculating ... Please Stand By ...' +
CAPTION ' Pause 3 with Gauge ' +
ICON WINDOWS +
OPTION GAUGE_VISIBLE ON +
|GAUGE_COLOR RED +
|GAUGE_INTERVAL 10 +
```



```

|MESSAGE_FONT_NAME VERDANA +
|MESSAGE_FONT_SIZE 10 +
|MESSAGE_FONT_COLOR WHITE +
|THEMENAME Steel Blue
-- Put your code, that takes a lot of time, here ...
-- Use CLS command to clear the PAUSE 3 with gauge
RETURN

```

14.47 Using the LAUNCH Command

Using R:BASE, the LAUNCH command is used to execute (or LAUNCH) another application or process, from within R:BASE eXtreme for Windows and higher. It designates which file to execute. This means that the FileName can be set to any file that has a corresponding association registered with Windows. For example, setting FileName to 'README.TXT' will invoke the Windows NotePad when launched. If a full path is not specified, the current search path is used.

The LAUNCH command can be used to launch programs, as well as files associated with programs.

If you are running R:BASE eXtreme on MS Windows 7, and if you have a file, such as Word document with built-in macros, you may use the following syntax to LAUNCH:

```

-- Example 01
LAUNCH 'winword|f:\temp\test.doc /mTestMacro|W'

```

If a folder path, file name, or parameter contains spaces, double quotes (") must be added.

```

-- Example 02
LAUNCH 'winword|"f:\temp word files\test.doc" "/mTestMacro"|W'

```

14.48 Using the New RUNRBA Command

Did you know that you can use the new RUNRBA command to Run R:BASE Application (.RBA) file in a command file or in an EEP?

A new RUNRBA command is now supported to specifically run an R:BASE Application (.RBA) file that is created specifically using the R:BASE Application Designer.

R:BASE will run an Application (.RBA) file like it is run within the Database Explorer, only from a command file or a Custom EEP.

```

-- Example
RUNRBA TestApp.rba

```

14.49 Using the PRNSETUP OPTION STANDARD Parameter

A custom PRNSETUP window was designed and introduced in R:BASE that allows developers to customize the Print Setup window with added support of R:BASE Themes, Caption, etc. In addition to OPTION parameters, this window allows the selection of Printer Name from the list of available Printers, Printer Properties, Print Range, and Copies.

However, if you wish your users to view and select extended printer options such as Paper Size, Paper Source (Tray), and Orientation (Portrait/Landscape), now you may use the new PRNSETUP OPTION STANDARD to provide such features.

```

-- Example 01 (To Display Custom R:BASE Print Window)
PRNSETUP

```

```

-- Example 02 (To Display Standard MS Print Window)

```

PRNSETUP OPTION STANDARD

14.50 Using the SCALEBY Property in R:BASE Application (.RBA)

Did you know that now you can use the SCALEBY property to dynamically adjust your application's main window or form based on a given screen resolution?

Syntax:

```
PROPERTY RBA_FORM SCALEBY 'nnnn'
```

Use this property to automatically scale an R:BASE Application (.RBA) form and all related controls to a specified screen pixel width. SCALEBY property modifies the Height and Width properties of the form and all controls placed on it, but it does not change the Top and Left properties; therefore, the size of the controls changes, but not the coordinates of the top-left corners of the controls. Thus, it preserves the position of all controls on the form.

If you have developed an R:BASE Application (.RBA) form that is 800x600 for the majority of users having a monitor with 800x600 screen resolution to fit the form on the entire available screen, but a few users of your application have a higher resolution (1024x768) and if you wish your 800x600 form to fill the entire screen of a monitor with a resolution of 1024x768, you'll need to use the RBA_FORM SCALEBY property as follows:

Example:

```
-- Application: On After Start Action
PROPERTY RBA_FORM SCALEBY '800'
RETURN
```

This will scale the original form designed as 800x600 to fill the entire screen even if the screen resolution is higher than 800x600, such as 1400x1050, 1280x800 or 1024x768.

Notes:

- The 'nnnn' value should exactly be the same as the original "form width" in order to scale proportionally on the monitors with higher resolution.
- Use the SCALEBY property as embedded custom EEP in "On After Start" Action for R:BASE Application (.RBA) file.
- To take advantage of this feature, make sure that you use the scaled true type fonts for all controls on that form.

14.51 Using the TOP and LEFT coordinates for PRNSETUP

There are times when you wish you could display the PRNSETUP window at a particular position on the desktop, instead of in the center of your screen.

Using the latest updates, now you can use the TOP and LEFT coordinates to achieve this goal.

Here's how:

```
PRNSETUP +
  OPTION CAPTION Select Printer +
  |TOP 10 +
  |LEFT 10 +
  |THEMENAME R:BASE Rocks!
RETURN
```

The PRNSETUP windows will be displayed at your desktop's top 10 and left 10 screen pixels, with a caption "Select Printer" and the theme of R:BASE Rocks!

You may change the TOP and LEFT coordinates, as well as choose from over 30 supported Themes, to suit your R:BASE application.

In addition to PRNSETUP, the OPTION for TOP and LEFT coordinates are also supported in the #WHERE, CHOOSE, DIALOG and PAUSE commands.

14.52 Using the Zoom Level Percent for PRINT

If you need to PRINT any report/label on SCREEN with a pre-defined Zoom Level, such as ZOOMPERCENT 55, you will need to precede the option with ZOOM_TYPE PERCENTAGE.

Example:

```
PRINT reportname WHERE whereclause ORDER BY orderbyclause +
OPTION SCREEN +
| WINDOW_STATE MAXIMIZED +
| ZOOM_TYPE PERCENTAGE +
| ZOOMPERCENT 55
```

14.53 Using Themes for the #WHERE Builder

Did you know that in addition to applying Themes to Forms, Pop-up Menus, CHOOSE, DIALOG and PAUSE commands, now you can also apply R:Themes for #WHERE Builder window also?

When using #WHERE IN tableview option of CHOOSE command, it opens the WHERE builder for you to enter conditions for the columns in the specified table or view; stores the entire WHERE clause in the variable varname.

-- Example 01: (Traditional Command)

```
CONNECT RRBYW19
CHOOSE vWhereClause FROM #WHERE IN Customer
RETURN
```

-- Example 02: (Using Razzmatazz theme for #WHERE)

```
CONNECT RRBYW19
CHOOSE vWhereClause FROM #WHERE IN Customer +
OPTION THEMENAME Razzmatazz
RETURN
```

-- Example 03: (Using R:BASE Rocks! theme for #WHERE)

```
CONNECT RRBYW19
CHOOSE vWhereClause FROM #WHERE IN Customer +
OPTION THEMENAME R:BASE Rocks!
RETURN
```

14.54 Using SET FEEDBACK

With SET FEEDBACK set to ON, R:BASE displays the number of rows processed and the elapsed time to completion in the new pause look-alike form at the center of the screen.

Syntax: SET FEEDBACK ON/OFF

Default: OFF

This setting displays processing results when either calculating, sorting, inserting or editing more than

100 rows.

Supported commands:

- ALTER TABLE
- COMPUTE
- CREATE INDEX
- CROSSTAB
- DELETE
- INSERT
- The data transfer for the JOIN of two tables
- LOAD
- The data transfer for a PROJECT command
- SELECT
- SORTing a huge record set
- the data transfer for a SUBTRACT of two tables
- TALLY
- The R:BASE UNION command

14.55 Using SET PROGRESS

With SET PROGRESS set to ON, did you know that you can display a cool progress bar display window to show the process progression when building indexes, packing or reloading a database?

Syntax: SET PROGRESS ON/OFF

Default: OFF

Example 01:

```
CONNECT RRBYW19
PACK INDEX
```

Example 02:

```
CONNECT RRBYW19
PACK TABLE Customer
PACK TABLE InvoiceHeader
PACK TABLE InvoiceDetail
```

Example 03:

```
CONNECT RRBYW19
RELOAD RRBYWNEW
```

Example 04:

```
DISCONNECT
SET MULTI OFF
PACK RRBYW19
SET MULTI ON
```

Example 05:

```
DISCONNECT
SET MULTI OFF
CONNECT RRBYW19
PACK
SET MULTI ON
```

14.56 Using Wild Cards with PROPERTY

Did you know that you can use the connected database wild card symbols (also known as MANY) to change the PROPERTY of all controls with matching Component IDs at once?

-- Example 01:

```
PROPERTY FontControl% VISIBLE 'FALSE'  
RETURN
```

-- Example 02:

```
PROPERTY %ontControl VISIBLE 'FALSE'  
RETURN
```

-- Example 03:

```
PROPERTY %ontCon% VISIBLE 'FALSE'  
RETURN
```

Imagine the possibilities!

14.57 Validating An Internet Connection

When designing and deploying R:BASE applications, you might have a cool R:BASE routine or Plugin to submit and/or retrieve data that requires an Internet connection. When your applications include tools to extract, transform and load data, they are mainly responsible for data transfer from your R:BASE application and the Internet to data warehouses.

However, not being connected to the Internet, or not having Internet access may result into -ERROR- message(s) and so forth. It would be nice to confirm a valid Internet connection before submitting or retrieving data or LAUNCHing a Web browser.

Did you know that now you can use a new GETPROPERTY command parameter to check whether or not the computer is connected to the Internet?

Here's how:

```
SET VAR vResult TEXT = NULL  
GETPROPERTY APPLICATION IsInternetConnected vResult
```

Resulting "vResult" variable will return the value of YES or NO.

Using this approach, you can now build higher intelligence into your R:BASE applications.

Part

XV

15 Functions

15.1 Capturing PRNSETUP INFO Results

PRNSETUP INFO retrieves a list of current printer settings like:

- Current Printer
- Paper Orientation
- Paper Size
- Paper Source
- Print Quality (DPI)
- Copies
- Color Mode
- Duplex Mode
- Collation

You may capture PRNSETUP INFO results using the "PRN" specific CVAL functions as follows:

- PRN_STATUS
- PRN_ORIENTATION
- PRN_SIZE
- PRN_SOURCE
- PRN_QUALITY
- PRN_COPIES
- PRN_COLORMODE
- PRN_DUPLEXMODE
- PRN_COLLATION

(CVAL('CURRENTPRINTER')) returns the current printer for the R:BASE session.

(CVAL('PRN_STATUS')) captures the printer status.

(CVAL('PRN_ORIENTATION')) captures the printer orientation.

(CVAL('PRN_SIZE')) captures the printer paper size.

(CVAL('PRN_SOURCE')) captures the printer paper source.

(CVAL('PRN_QUALITY')) captures the printer print quality (DPI).

(CVAL('PRN_COPIES')) captures the printer copy count.

(CVAL('PRN_COLORMODE')) captures the printer color mode.

(CVAL('PRN_DUPLEXMODE')) captures the printer duplex mode.

(CVAL('PRN_COLLATION')) captures the printer collation.

15.2 Defining First and Last Date of the Current Month

There are times when you may need to provide your users with a pre-defined first and last date of the month for query and/or reporting purposes.

Did you know that you can dynamically define the first and last date of the current month by using the following two lines of code:

```
SET VAR vStartDate DATE = (RDATE((IMON(.#DATE)),1,(IYR4(.#DATE))))
SET VAR vEndDate DATE = ((ADDMON(.vStartDate,1))-1)
```

The two lines of code above will always calculate the first and last date of the current month regardless of the current date.

15.3 Finding PORTS on Your Workstation

Did you know that you can find the available ports on your computer?

You can find all available ports using the `*new*` (`CVAL('PORTS')`) function.

Here's how:

1. Start R:BASE
2. At the R> prompt, type:

```
SET VAR vAvailablePorts TEXT = NULL
SET VAR vAvailablePorts = (CVAL('PORTS'))
SHOW VAR vAvailablePorts
```

The variable `vAvailablePorts` will return a text string with a list of all ports that are available on that workstation. Each item in the list will be separated by a comma (or the database character settings for comma).

15.4 Finding Relative Word Position in a Sentence

Did you know that you can find the relative word position in a sentence?

Syntax:

```
(RWP('string','actualword',occurrence))
```

The RWP function in R:BASE returns the position of a specified word and occurrence within a TEXT, NOTE or VARCHAR string.

```
-- Example 01:
SET VAR vRWP1 INTEGER = +
(RWP('What is the next major version of R:BASE on horizon','horizon',1))
Variable vRWP1 will return the value of 10.
```

```
-- Example 02:
SET VAR vRWP2 INTEGER = +
(RWP('How many years has it been in development','many',1))
Variable vRWP2 will return the value of 2.
```

```
-- Example 03:
SET VAR vRWP3 INTEGER = (RWP('What will the product name be','name',1))
Variable vRWP3 will return the value of 5.
```

```
-- Example 04:
SET VAR vRWP4 INTEGER = (RWP('When will the big announcement
be','when',1))
Variable vRWP4 will return the value of 1.
```

```
-- Example 05:
SET VAR vRWP5 INTEGER = +
(RWP('We have shocked the world and we will do it again!','we',2))
Variable vRWP5 will return the value of 7.
```

```
-- Example 06:
```



```
SET VAR vRWP6 INTEGER = (RWP('Remember, nothing is
impossible', 'nothing', 1))
Variable vRWP6 will return the value of 2.
```

15.5 Finding Table/View Locks Programmatically

Did you know that you can find the table or view locks using the new (GETVAL('GetLock','tableviewname')) function in R:BASE?

Here's how:

```
SET VARIABLE vCheckLock TEXT = NULL
SET VARIABLE vCheckLock = (GETVAL('GetLock', 'tableviewname'))
```

Where:

GetLock is the first required parameter and the tableviewname is the name of table or view.

The variable *vCheckLock* will return the text value of ON or OFF.

15.6 Finding the Data Type for a given Column or Variable

Did you know that you can find the data type for a given column name or variable by using the undocumented CVTYPE function?

Here's how:

Syntax:

```
(CVTYPE('colvarname', flag))
```

Returns the data type for a given column or variable name. To return the data type for a given column, a zero "0" flag must be used. To return the data type for a given variable, the one "1" flag must be used.

1. Start R:BASE for DOS or Windows
2. After CONNECTing to ConComp or RRBYW19, try the following two examples at the R> prompt:

Example 01.

```
SET VAR vCustIDType TEXT = (CVTYPE('CustID', 0))
SET VAR vEmpCity TEXT = (CVTYPE('EmpCity', 0))

SHOW VARIABLES

vCustIDType = INTEGER TEXT
vEmpCity = TEXT,16 TEXT
```

Example 02.

```
SET VAR vCustIDType TEXT = (CVTYPE('CustID', 0))
SET VAR vVarInquiry TEXT = (CVTYPE('vCustIDType', 1))

SHOW VARIABLES

vCustIDType = INTEGER TEXT
vVarInquiry = TEXT,7 TEXT
```

Notes:

- When using the zero flag to return column data types, you must be CONNECTed to a database.
- When returning a TEXT data type, the length is included and is separated with a comma. When returning a NUMERIC data type, the precision and scale are separated with commas.

15.7 Finding the INTEGER value for a given TEXT String

Did you know that you can find the INTEGER value for a given TEXT string by using the ISTR function of R:BASE?

Here's how:

Syntax:

```
( ISTR( 'textstring', position) )
```

Returns the corresponding integer value.

This function converts a single character, which you specify within a string by position, returning its corresponding ASCII Character Chart Decimal value.

In the following example, the INTEGER value of *vDecimalValue* is 65 for the capital letter A.

Example 01:

Start R:BASE for DOS or R:BASE for Windows at the R> Prompt:

```
SET VARIABLE vDecimalValue = ( ISTR( 'R:BASE Rocks!', 4) )

SHOW VARIABLE

vDecimalValue = 65 INTEGER
```

15.8 Managing Application Security Using New CVAL Functions

Traditionally, when managing database/application security and access level, you may validate and provide database and application access by implementing various R:BASE (CVAL('keyword')) functions as follows:

- (CVAL('COMPUTER')) Returns the name of your computer
- (CVAL('NETUSER')) Returns the currently logged-in network user name
- (CVAL('USERDOMAIN')) Returns the Name of Logged-In Network Domain
- (CVAL('USERID')) Returns the Windows User ID
- (CVAL('WINAUTH')) Returns the value of the WINAUTH operating condition. This function is specific to R:BASE X Enterprise.

Did you know that now you may provide added database/application security and access to windows logged-in user based upon MS Windows "Global Groups" and "Local Groups"?

- (CVAL('NETGROUP')) Returns a list of global groups to which the current logged-in user belongs
- (CVAL('NETLOCALGROUP')) Returns a list of local groups to which the current logged-in user belongs

Using these CVAL functions, now you may provide access based on your network global or local groups as compared to providing application access based on individual (CVAL('NetUser')).

15.9 Managing DLCALL Functions

Did you know that you can:

1. Declare up to 200 functions using the DLCALL function
2. Use CHKFUNC function to check DLL function that exists or not

Example:

```
SET VAR v1 = (CHKFUNC('function_name'))
```

Returns 1 if function exists or 0 if not.

3. Use DELFUNC function to delete DLL function declaration

Example:

```
SET VAR v1 = (DELFUNC('function_name'))
```

Returns 1 if function was deleted or 0 if not.

4. Use new LIST FUNCTIONS command to list declared functions

LIST FUNCTIONS command will list all declared DLL functions

15.10 Rounding Down with RNDDOWN Function

Did you know that you can use R:BASE to round down numbers to a specific decimal place?

The RNDDOWN function returns a number rounded down to a specified number of digits.

Syntax:

```
(RNDDOWN(number,digits))
```

Where:

number is the number to round down

digits is the number of digits to round the number down to

Remarks:

- RNDDOWN behaves like ROUND, except that it always rounds a number down.
- If digits is greater than 0 (zero), then number is rounded down to the specified number of decimal places.
- If digits is 0, then number is rounded down to the nearest integer.
- If digits is less than 0, then number is rounded down to the left of the decimal point.

Examples:

--Example 01:

```
SET VAR v1 DOUBLE = (RNDDOWN(662.79,0))
SHOW VAR v1
662
```

--Example 02:

```
SET VAR v3 DOUBLE = (RNDDOWN(54.1, -1))
SHOW VAR v3
50
```

```
--Example 03:
SET VAR v5 DOUBLE = (RNDDOWN(-23.67, 1))
SHOW VAR v5
-23.6
```

15.11 Understanding the New IFF Logical Function

Did you know that a new "IFF" logical function has been added to handle working with conditions to return a value.

Syntax:

```
(IFF((condition), arg1, arg2))
```

If the condition is met, then the value of arg1 is returned. If condition is not met, then the value of arg2 is returned.

The condition must follow the syntax specifications as set with IF...ENDIF command structure.

The condition must list a set of conditions that combine to form a statement that is either true or false.

Conditions can be combined with the connecting operators AND, OR, AND NOT, and OR NOT.

Conditions using CURRENCY, the value MUST match database CURRENCY SYMBOL settings.

It is important to note that the condition needs to be a single item, which is why quotes are used in the examples below.

The data types of arg1 and arg2 must match.

```
-- Example 01: (Comparison of Variables)
SET VAR v1 INTEGER = 1
SET VAR v2 INTEGER = 2
SET VAR v3 TEXT = NULL
SET VAR v3 = (IFF('.v1<.v2','First','Second'))
SHOW VAR v3
First
```

```
-- Example 02: (Comparing Logic)
SET VAR v1 TEXT = NULL
SET VAR v1 = (IFF('1 < 2','Yes','No'))
SHOW VAR v1
Yes
```

```
-- Example 03: (Using IF Command)
SET VAR v1 INTEGER = 1
SET VAR v2 INTEGER = 2
IF v1 < .v2 THEN
PAUSE 2 USING 'Smaller' +
CAPTION 'New IFF Function' ICON INFO +
BUTTON 'Press any key to continue ..' +
OPTION MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_COLOR GREEN +
|MESSAGE_FONT_SIZE 11 +
|THEMENAME Longhorn
ELSE
PAUSE 2 USING 'Bigger' +
```

```

CAPTION 'New IFF Function' ICON INFO +
BUTTON 'Press any key to continue ..' +
OPTION MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_COLOR GREEN +
|MESSAGE_FONT_SIZE 11 +
|THEMENAME Longhorn
ENDIF
CLEAR VARIABLES v1,v2
RETURN

```

-- Example 04: (Using WHILE Command)

```

SET VAR v1 INTEGER = 1
SET VAR v2 INTEGER = 2
SET VAR vPause2Message TEXT = NULL
WHILE v1 < v2 THEN
SET VAR v1 = (.v1 + 1)
SET VAR vPause2Message = (CTXT(.v1))
PAUSE 2 USING .vPause2Message +
CAPTION 'New IFF Function' ICON INFO +
BUTTON 'Press any key to continue ..' +
OPTION MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_COLOR GREEN +
|MESSAGE_FONT_SIZE 11 +
|THEMENAME Longhorn
ENDWHILE
CLEAR VARIABLES v1,v2,vPause2Message
RETURN

```

-- Example 05: (SELECT and Display Feedback on Data)

```

CONNECT RRBYW19
SELECT NetAmount, (IFF('NetAmount>$100000','Good','Needs Improvement'))=30
+
AS Status FROM SalesBonus

NetAmount Status
-----
$176,000.00 Good
$87,500.00 Needs Improvement
$22,500.00 Needs Improvement
$40,500.00 Needs Improvement
$76,800.00 Needs Improvement
$36,625.00 Needs Improvement
$56,250.00 Needs Improvement
$152,250.00 Good
$108,750.00 Good
$78,750.00 Needs Improvement
$27,000.00 Needs Improvement
$9,500.00 Needs Improvement
$210,625.00 Good

```

15.12 Using Additional Logical Functions

R:BASE's Logical Functions can be used to introduce decision making into your application. Traditional Logical Functions include:

- IFEQ
- IFEXISTS
- IFGT
- IFLT
- IFNULL

(IFEQ(arg1,arg2,arg3,arg4))

If arg1 and arg2 are equal, IFEQ returns the value of arg3. If arg1 and arg2 are not equal, IFEQ returns the value of arg4.

(IFEXISTS(arg1,arg2,arg3))

If arg1 contains a value, IFEXISTS returns the value of arg2. If arg1 is null, then the value of arg3 is returned.

(IFGT(arg1,arg2,arg3,arg4))

If arg1 is greater than arg2, IFGT returns the value of arg3. If arg1 is less than arg2, IFGT returns the value of arg4

(IFLT(arg1,arg2,arg3,arg4))

If arg1 is less than arg2, IFLT returns the value of arg3. If arg1 is greater than arg2, IFLT returns the value of arg4.

(IFNULL(arg1,arg2,arg3))

If arg1 is null, then the value of arg2 is returned. If arg1 is not null, then the value of arg3 is returned.

Did you know that the latest version and update of R:BASE eXtreme 9.5 include the following additional Logical Functions?

- IFGE
- IFLE
- IFNE

(IFGE(arg1,arg2,arg3,arg4))

If arg1 is greater than or equal to arg2, IFGE returns the value of arg3. If arg1 is less than arg2, IFGE returns the value of arg4.

(IFLE(arg1,arg2,arg3,arg4))

If arg1 is less than or equal to arg2, IFLE returns the value of arg3. If arg1 is greater than arg2, IFLE returns the value of arg4.

(IFNE(arg1,arg2,arg3,arg4))

If arg1 and arg2 are not equal, IFNE returns the value of arg3. If arg1 and arg2 are equal, IFNE returns the value of arg4.

15.13 Using Enhanced ICAP2 Function

The ICAP2 Function in R:BASE converts the text string with an initial capital letter on each word. This is also known as Word Case.

In the following example, the value of vICAP2 is John Smith.

```
SET VAR vICAP2 TEXT = (ICAP2('john smith'))
```

In the following example, the value of vHyfnTxt2 is Daniel Day-Lewis.

```
SET VAR vHyfnTxt2 TEXT = (ICAP2('daniel day-lewis'))
```

Did you know that the R:BASE now includes the support for apostrophe as well?

In the following example, the value of vName2 is Brian O'Rielly.

```
SET VAR vName1 TEXT = ('BRIAN O'RIELLY')
SET VAR vName2 TEXT = (ICAP2(.vName1))
```

This new enhancement will help those who wish to see consistent results when using the ICAP2 function to display Word Case in so many cases, such as List Views, Mail Merge Letters, Mailing/Name Labels, Inscription applications, and more ...

15.14 Using IFWINDOW

Did you know that you can use the IFWINDOW function to see if the form is open or not, WITHOUT specifying the AS alias option of ENTER or EDIT USING formname?

```
(IFWINDOW(FormComponentID))
```

will return 1 when a form with the Component ID is open, 0 if not.

All you need to do is assign a Component ID to the form.

15.15 Using IINFO

There is a IINFO function in R:BASE which returns information about tables, columns, or indexes by reading internal bitmask flags. The function requires the ID number for the table, column, or index. This ID number can be obtained from the system tables SYS_TABLES, SYS_COLUMNS, or SYS_INDEXES, respectively. IINFO returns 0 if FALSE, or the number in argument 3 if TRUE.

Syntax:

```
(IINFO(flagtype, id, bitmask))
```

Where:

flagtype - specifies if the info returned is for table flags, column flags, column flags for server tables, or index flags

id - specifies the ID number from the system tables for the table ID, column ID, or index ID

bitmask - species the flag in the system table

Remarks:

. The values for arg1, arg2, and arg3 must be non-null integers, even if a particular argument is not needed for that case.

. IINFO returns 0 if FALSE, or the bitmask number in parameter 3 if TRUE (flags 4-7).

Flags:

Info	Flag Type	ID	Bitmask	Description
Row ID	0	0	0	Returns the rowid of the current row.
Minimum Data Type Scale	1	integer	0	Returns the minimum scale for the data type.

Maximum Data Type Scale	2	integer	0	Returns the maximum scale for the data type
Table Cascade Flag	3	table ID	0	Returns cascade flag for a table
Column Flags	4	column ID	1	Returns bitmask value if is an autonumber column
exists for column	4	column ID	2	Returns bitmask value if a comment exists for column
a default value	4	column ID	4	Returns bitmask value if column has a default value
temporary	4	column ID	8	Returns bitmask value if column is temporary
an index	4	column ID	16	Returns bitmask value if column has an index
USER default	4	column ID	32	Returns bitmask value if contains a USER default
Not NULL flag	4	column ID	64	Returns bitmask value if contains a Not NULL flag
primary key or unique key	4	column ID	128	Returns bitmask value if is a primary key or unique key
Column Flags	5	column ID	1	Returns bitmask value if column is an optimal row qualifier
(Server Tables) column is read only	5	column ID	2	Returns bitmask value if server column is read only
column is autonumbered	5	column ID	4	Returns bitmask value if server column is autonumbered
column row version qualifier	5	column ID	8	Returns bitmask value if server column row version qualifier
Table Flags	6	table ID	1	Returns bitmask value if comment exists for table
primary key	6	table ID	2	Returns bitmask value if table has a primary key
foreign key	6	table ID	4	Returns bitmask value if table has a foreign key
an autonumbered column	6	table ID	8	Returns bitmask value if table has an autonumbered column
default column	6	table ID	16	Returns bitmask value if table has a default column
readonly (dBASE)	6	table ID	32	Returns bitmask value if table is readonly (dBASE)
temporary	6	table ID	64	Returns bitmask value if table is temporary
referenced key	6	table ID	128	Returns bitmask value if table has a referenced key
Not NULL column	6	table ID	256	Returns bitmask value if table has a Not NULL column
unique key	6	table ID	512	Returns bitmask value if table has a unique key
	6	table ID	1024	Returns bitmask value if table has a

column with a data type				greater than 10
VARBIT/VARCHAR column	6	table ID	2048	Returns bitmask value if table has a
flag updates and deletes	6	table ID	4096	Returns bitmask value if a cascade
keys				through all primary keys and unique
table has column aliases	6	table ID	8192	Returns bitmask value if server
least one trigger defined	6	table ID	16384	Returns bitmask value if there is at
system view which created	6	table ID	32768	Returns bitmask value if relation as
				during multiple inner joins
Index Flags	7	index ID	3	Returns bitmask value for the
constraint type:				
dBase index	7	index ID	4	Returns bitmask value if this is a
unique index	7	index ID	8	Returns bitmask value if this is a
temporary	7	index ID	16	Returns bitmask value if index is
referenced key	7	index ID	32	Returns bitmask value if this is a
case sensitive index	7	index ID	64	Returns bitmask value if this is a
Foreign Index	7	index ID	128	Returns bitmask value if the is a

Examples:

```
-- Example 01:
-- Using flag type 0 for the Titles table in the RRBYW18 database
SELECT EmpTID,EmpTitle,(IINFO(0,0,0)) FROM Titles
```

EmpTID	EmpTitle	(IINFO(0,0
-----	-----	-----
1	Office Manager	524289
2	Receptionist	524341
3	Sales Clerk	524393
4	Director Marketing	524445
5	Director Corporate Sales	524497
6	Director Government Sales	524549
7	Manager Support & Services	524601
8	Outside Sales	524653

```
-- Example 02:
-- Returns minimum and maximum data type scales using IINFO and (CVAL('ROWCOUNT')) with the
RRBYW18 database.
-- The values for currency will vary based upon the current CURRENCY DIGITS setting.
SELECT (CVAL('ROWCOUNT')) AS SYS_TYPE, SYS_TYPE_NAME=18, +
(IINFO(1, (INT(CVAL('ROWCOUNT'))), 0)) AS SYS_MIN, +
(IINFO(2, (INT(CVAL('ROWCOUNT'))), 0)) AS SYS_MAX +
FROM SYS_TYPES
```

Here is what it generates:

SYS_TYPE	SYS_TYPE_NAME	SYS_MIN	SYS_MAX
1	CURRENCY		2
2	VARBIT	-0-	-0-
3	BITNOTE	-0-	-0-
4	BIT	-0-	-0-
5	VARCHAR	-0-	-0-
6	BIGNUM	-0-	-0-
7	BSTR	-0-	-0-
8	GUID	-0-	-0-
9	TEXT	-0-	-0-
10	NUMERIC		0 15
11	INTEGER		0 0
12	REAL	-0-	-0-
13	DOUBLE	-0-	-0-
14	DATE	-0-	-0-
15	TIME		0 3
16	DATETIME		0 3
17	NOTE	-0-	-0-

-- Example 03:

-- Displays tables with Cascade within the RRBYW18 database.

-- Note that the tables with 1 for the cascade value are tables

-- with primary keys that cascade to tables with foreign keys.

```
SELECT SYS_TABLE_NAME=20, +
SYS_TABLE_ID, +
(IINFO(3,SYS_TABLE_ID,0)) AS SYS_CASCADE +
FROM SYS_TABLES WHERE SYS_TABLE_TYPE = 'TABLE'
```

SYS_TABLE_NAME	SYS_TABLE_	SYS_CASCAD
Customer	29	1
CompUsed	30	0
SalesBonus	31	0
PaymentTerms	32	0
Contact	33	0
ProdLocation	34	0
Levels	35	0
Component	36	0
Product	37	0
SecurityTable	38	0
InvoiceHeader	39	0
PrintOptions	40	0
InvoiceDetail	41	0
LicenseInformation	43	0
Titles	44	1
Employee	45	1
StateAbr	46	0
FormTable	47	0
BonusRate	48	0
TestNote	49	0
RThemes_eXtreme	50	0

```
ContactCallNotes          51          0
tempemployee              74          0
```

-- Example 04:

-- Checks that the CustState colun in the Customer table is indexed

```
SET VAR vCustStateHasIndex = (IINFO(4,191,16))
SHOW VAR vCustStateHasIndex
      16
```

-- Example 05:

-- Checks that the Employee table has a primary key

```
SET VAR vEmployeeHasPK = (IINFO(6,45,2))
SHOW VAR vEmployeeHasPK
      2
```

-- Example 06:

-- Checks that the Component table has a referenced key.

```
SET VAR vComponentRef = (IINFO(6,39,128))
SHOW VAR vComponentRef
      128
```

-- Example 07:

-- Returns the constraint type for the EmpID in the SalesBonus table

-- (0 = index, 1 = foreign key, 2 = primary key, 3 = unique key)

```
SET VAR vIsForeignKey = (IINFO(7,42,3))
SHOW VAR vIsForeignKey
      1
```

15.16 Using CHKCUR

If you need to know whether a CURSOR is already DECLARED or DROPPed, now you may use the CHKCUR function to find the current status of a given cursor name.

Here's how:

```
SET VARIABLE vCheckCuror = (CHKCUR('cursorname'))
```

The CHKCUR Function checks to see if a cursor is declared and is not dropped. The function returns an integer value of 1 if the cursor exists and is not dropped, and 0 if it is not declared or dropped.

The variable vCheckCursor will return a value of 1 or 0.

15.17 Using CHKVAR

Have you ever had the need to find whether a required VARIABLE is declared before you use that variable in an expression or routine to avoid code breaks or a sequence of -ERROR- messages?

Now you can avoid those moments by simply verifying the existence of a variable using the CHKVAR function, which checks to see if a variable name exists.

Here's how:

```
SET VARIABLE vCheckVar = (CHKVAR('varname'))
```

The function returns an integer value of 1 if the variable name exists or is declared, and 0 if it is not defined.

The variable vCheckVar will return a value of 1 or 0.

No more workarounds, suppressing error messages or even lengthy code to evaluate the existence of a variable!

15.18 Using SLOCP

The SLOCP Function is used to locate the exact position of a given string and occurrence in a TEXT, NOTE or VARCHAR value, returning the position if the string is found or 0 if it is not found. Using -1 as the third parameter will return the LAST occurrence.

Syntax:

```
(SLOCP(TextNoteVarcharValue,string,occurrence))
```

Example 01:

```
SET VARIABLE v1 VARCHAR = 'ABCDEABC_AB'  
SET VARIABLE vSloc1 = (SLOCP(.v1,'AB',1))
```

The resulting variable *vSloc1* will return the integer value of 1.

Example 02:

```
SET VARIABLE v1 VARCHAR = 'ABCDEABC_AB'  
SET VARIABLE vSloc2 = (SLOCP(.v1,'AB',2))
```

Resulting variable *vSloc2* will return the integer value of 6.

The above two examples will locate the exact position of a given string and occurrence. However, if you wish to find the LAST occurrence in the value of your string, use the "-1" parameter. Here's how:

Example 03:

```
SET VARIABLE v1 VARCHAR = 'ABCDEABC_AB'  
SET VARIABLE vSlocLast = (SLOCP(.v1,'AB',-1))
```

Resulting variable *vSlocLast* will return the integer value of 10.

15.19 Using SLOCI Function to Locate the Number of String Instances

Did you know that you can locate the number of instances a string appears in a TEXT, NOTE, or VARCHAR value using the new SLOCI Function?

Syntax:

```
(SLOCI(TextNoteVarcharValue,string,arg))
```

New SLOCI function will return the INTEGER value for the number of instances a string appears in a TEXT, NOTE, or VARCHAR value.

The argument parameter determines whether the search on the string is case sensitive, where "0" is not case sensitive while "1" is case sensitive.

Example 01:

```
SET VAR vProduct1 TEXT = (SLOCI('R:BASE Single Seat: Upgrade',':',0))  
SHOW VAR  
vProduct1 2 INTEGER
```

Example 02:

```
SET VAR vProduct2 TEXT = (SLOCI('Single Seat License','S',1))
SHOW VAR
vProduct2 2 INTEGER
```

Example 03:

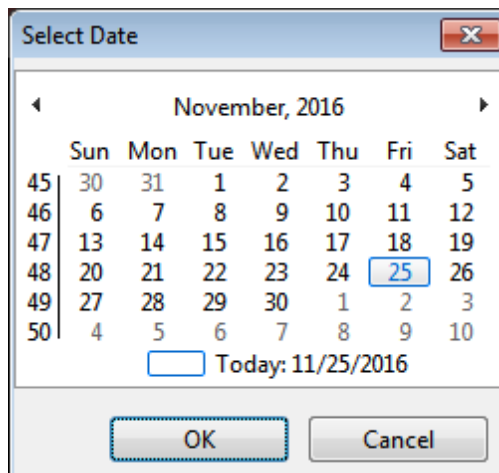
```
SET VAR vProduct3 TEXT = (SLOCI('Single Seat License','S',0))
SHOW VAR
vProduct3 3 INTEGER
```

Have fun implementing this new Function in your R:BASE applications!

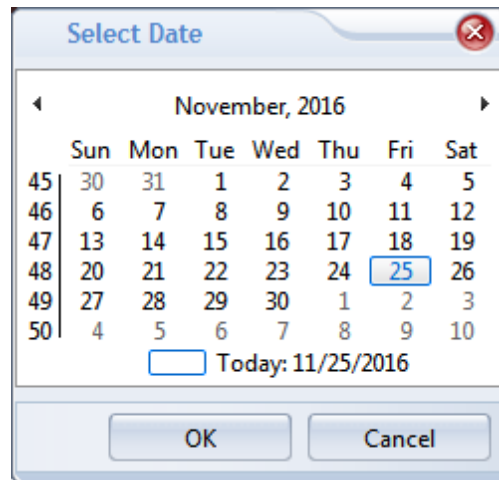
15.20 Using the Enhanced (GETDATE(' ')) Options

Traditionally, (GETDATE(' ')) will bring up the Windows GUI Calendar with today's date circled in red. Either you can click on [OK] to accept the circled date or click on any other date on the calendar using the current month or scrolling months.

```
-- Example 01 (Traditional Calendar):
SET VAR vDate DATE = NULL
SET VAR vDate = (GETDATE('Select Date'))
RETURN
```

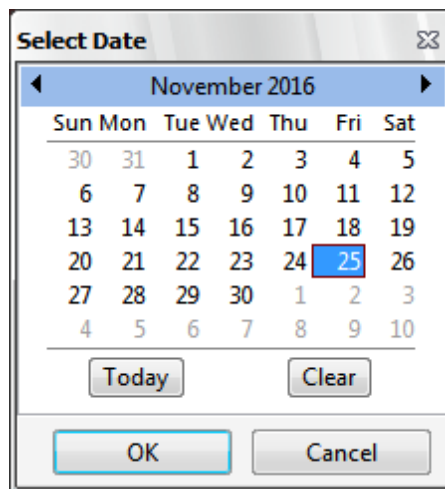


```
-- Example 02 (Traditional Calendar with R:BASE Themes):
SET VAR vDate DATE = NULL
SET VAR vDate = (GETDATE('Select Date|THEMENAME Soft Blue'))
RETURN
```



Did you know that now you can use the new "CALENDAR_TYPE ENHANCED" option to display an enhanced calendar?

```
-- Example 03 (Enhanced Calendar with R:BASE Theme):
SET VAR vDate DATE = NULL
SET VAR vDate = +
(GETDATE('Select Date|CALENDAR_TYPE ENHANCED|THEMENAME Vista CG'))
RETURN
```



Using the enhanced calendar, additional [Clear] button will clear the selection and use today's date as default. Clicking on [OK] will populate the selected date into a variable, such as vDate.

Clicking on [Cancel] will ignore to populate any value into a variable, such as vDate. Pre-Defined value, if used, will not be overwritten.

```
-- Example 04 (Dynamic Approach using Global Variables)
SET VAR vDate DATE = NULL
SET VAR vCaption TEXT = 'Select Date'
SET VAR vCalendarType TEXT = 'ENHANCED'
SET VAR vThemeName TEXT = 'XP Grey Scale'
SET VAR vQuotes = (CVAL('QUOTES'))
```

```

SET VAR vString TEXT = +
('SET VAR vDate = (GETDATE('+vQuotes+.vCaption+ +
'|CALENDAR_TYPE'+.vCalendarType+ +
'|THEMENAME'&.vThemeName+.vQuotes+'))')
&vString
CLEAR VARIABLE vCaption,vCalendarType,vThemeName,vQuotes,vString
RETURN

```

15.21 Using the FORMAT Function with .#NOW

Technically, you can use FORMAT anywhere that you can use a function. The result of the FORMAT function is always text. In the syntax for this function, value is the value you want to be displayed in a particular format; it can be a column, variable, or a constant value. 'Picture-format' is the picture format you establish.

In a command file, the FORMAT function can be useful in several ways; such as, aligning decimals, punctuating long numbers, formatting currency, and formatting text for CHOOSE, DIALOG, and PAUSE commands.

A beneficial example using FORMAT, for creating dynamic and unique file names based on the current date and time, would be useful to any user/developer that requires the ability to make an unattended timely backup of your live database.

The following FORMAT function examples can be used to create a dynamic file name based on .#DATE, .#TIME or .#NOW.

Let's say if you are already connected to the RRBYW19 database and would like to implement a routine to backup the entire database (even in a multi-user environment) on demand.

Here's a simple routine to create a unique file name:

```

-- Example 01: (Using .#DATE and .#TIME as two separate variables)
CLS
CLEAR VARIABLES vDateTxt,vTimeTxt,vFileName
SET VAR vDateTxt TEXT = NULL
SET VAR vTimeTxt TEXT = NULL
SET VAR vFileName TEXT = NULL
SET VAR vDateTxt = (FORMAT(.#DATE,'MMDDYYYY'))
SET VAR vTimeTxt = (FORMAT(.#TIME,'HHMM'))
SET VAR vFileName = +
((CVAL('DATABASE'))+'_'+vDateTxt+'_'+vTimeTxt+'.BKP')
CLEAR VARIABLES vDateTxt,vTimeTxt
RETURN
-- vFileName will return the text variable as:
-- RRBYW19_12302017_1630.BKP
-- Database Name, Date and Time will vary on your end

-- Example 02: (Using .#DATE and .#TIME as one variable in expression)
CLS
CLEAR VARIABLES vFileName
SET VAR vFileName TEXT = NULL
SET VAR vFileName = +
((CVAL('DATABASE'))+'_'+(FORMAT(.#DATE,'MMDDYYYY')))+ +
'_'+(FORMAT(.#TIME,'HHMM'))+'.BKP')
RETURN
-- vFileName will return the text variable as:
-- RRBYW19_12302017_1630.BKP
-- Database Name, Date and Time will vary on your end

```

```
-- Example 03: (Using .#NOW) - Short and Swift
CLS
CLEAR VARIABLES vFileName
SET VAR vFileName TEXT = NULL
SET VAR vFileName = +
((CVAL('DATABASE'))+'-'+(FORMAT(.#NOW,'MMDDYYYY_HHNN'))+'.BKP')
RETURN
-- Notice the "NN" for minutes when using .#NOW (not a typo)
-- vFileName will return the text variable as:
-- RRBYW19_12302017_1630.BKP
-- Database Name, Date and Time will vary on your end
```

Now that you have successfully created a unique file name, you may use the following routine to make a backup of your live database, on demand!

Here's how:

```
OUTPUT .vFileName
UNLOAD ALL
OUTPUT SCREEN
```

The resulting two files will be created in the same folder, unless a different folder name is specified:

1. RRBYW19_12302017_1630.BKP
2. RRBYW19_12302017_1630.LOB

15.22 Using the LISTOF Function for Large Data

Did you know that you can use VARCHAR data type to retrieve large dataset when using the (LISTOF(colname)) function?

Syntax:

```
SELECT (LISTOF(colname)) IN TableViewName

SELECT (LISTOF(colname)) INTO varname INDIC ivarname FROM tblviewname
```

Note: Use WHERE clause to limit the resulting data, if necessary.

The (LISTOF(colname)) function creates a text string of the matched values separated by the current comma delimiter character.

The LISTOF function can be used with the "SELECT ... INTO ..." to populate a variable with a list of values which can then be used in a CHOOSE command with the #LIST option. It can also be used in Forms, Reports or Labels to look up values from multiple rows.

By default, the (LISTOF(colname)) function returns a NOTE data type and notes are limited to 4K. However, if you need to retrieve a large data set to build a dynamic CHOOSE command, Dynamic ListView, or even Dynamic ListBox, now you can use the VARCHAR data type to achieve your goal.

Example 01:

```
CONNECT RRBYW19
SET VARIABLE vResult VARCHAR = NULL
SELECT (LISTOF(CompDesc)) IN Component
```

Example 02:

```
CONNECT RRBYW19
```



```
SET VARIABLE vResult VARCHAR = NULL
SELECT (LISTOF(CompDesc)) IN Component WHERE CompDesc CONTAINS 'Box'
```

Example 03:

```
CONNECT RRBYW19
SET VARIABLE vResult VARCHAR = NULL
SELECT (LISTOF(CompDesc)) INTO vResult INDIC ivResult FROM Component
SHOW VARIABLE vResult
```

Example 04:

```
CONNECT RRBYW19
SET VARIABLE vResult VARCHAR = NULL
SELECT (LISTOF(CompDesc)) INTO vResult INDIC ivResult FROM +
  Component WHERE CompDesc CONTAINS 'Box'
SHOW VARIABLE vResult
```

15.23 Using TINFO

There is a TINFO function in R:BASE which returns access right information for the current user.

```
(TINFO(arg1,arg2,arg3))
```

Where:

- arg1 is zero (0), which specifies to show table permissions
- arg2 is the system table ID (SYS_TABLE_ID) value (can be located in the SYS_TABLES system table)
- arg3 can be:
 - a) A specific system column ID (SYS_COLUMN_ID) in that table
 - b) A value of 0 which means show permissions that apply to all the columns in that table
 - c) A value of -1 which means show permissions that apply to any of the columns in that table

Remarks:

- TINFO returns a comma delimited string of the access rights.
- In most instances table and column privileges are identical. Cases where they are not are autonumber column (no INSERT), computed columns, (no INSERT or UPDATE), and where the UPDATE privilege was granted to specific columns only.
- Permissions are returned based upon the current USER.

Examples:

The following example is based upon a database configured with an OWNER and users, where limited permissions are supplied to a user Jim for the Component table.

While connected to the database as the OWNER, the system table ID and system column ID can be found for the Component table and CompDesc column.

```
SELECT SYS_TABLE_ID FROM SYS_TABLES WHERE SYS_TABLE_NAME = 'Component'
SYS_TABLE_ID
-----
29
R>SELECT SYS_COLUMN_ID FROM SYS_COLUMNS WHERE SYS_COLUMN_NAME =
'CompDesc'
SYS_COLUMN_ID
-----
188
R>SET USER JIM JIM999
```

Example 01:

```
-- Permissions specifically for the column CompDesc
R>SET VAR vUserInfoColumn = (TINFO(0,29,188))
R>SHOW VAR vUserInfoColumn
SELECT, UPDATE
```

Example 02:

```
-- Permissions for all columns.
-- Only SELECT is returned since that is the only permission applied to
all columns.
R>SET VAR vUserInfoAllColumns = (TINFO(0,29,0))
R>SHOW VAR vUserInfoAllColumns
SELECT
```

Example 03:

```
-- Permissions for any column.
-- Both SELECT and UPDATE are returned since there are some columns with
both of those permissions.
R>SET VAR vUserInfoAnyColumn = (TINFO(0,29,-1))
R>SHOW VAR vUserInfoAnyColumn
SELECT, UPDATE
```

Part

XVI

16 R:Compiler

16.1 Customizing the Load Window in Compiled Applications

When compiling your application, you have the option (check box) to "Show Loading Window" which permits the ability to display a progress window while the program is loading. The available field is used to enter a custom message for the progress dialog window.

With this feature available to the compiled program, there are additional PROPERTY command parameters which you can use to display customized load window:

```
PROPERTY LOADWINDOW SHOWINDICATOR 'TRUE'  
PROPERTY LOADWINDOW PROGRESS nnn (0-100)  
PROPERTY LOADWINDOW CAPTION 'Your Caption'  
PROPERTY LOADWINDOW TITLE 'Your Message Here'  
PROPERTY LOADWINDOW BACKGROUND_COLOR 'Color'  
PROPERTY LOADWINDOW TITLE_FONT_NAME 'Font Name'  
PROPERTY LOADWINDOW TITLE_FONT_SIZE nn  
PROPERTY LOADWINDOW TITLE_FONT_BOLD ON/OFF  
PROPERTY LOADWINDOW TITLE_FONT_ITALIC ON/OFF  
PROPERTY LOADWINDOW TITLE_FONT_UNDERLINE ON/OFF  
PROPERTY LOADWINDOW TITLE_FONT_STRIKE_OUT ON/OFF  
PROPERTY LOADWINDOW TITLE_FONT_COLOR 'Color'  
PROPERTY LOADWINDOW CLOSE 'TRUE'
```

* The above PROPERTY commands are supported for R:Compiler ONLY.

01. While customizing your R:Compiler project, make sure to enable the option for "Show Loading Windows"

General Information

R:BASE Startup File Name
F:\RBGX_Demo\RRBYW19\Original\RRBYW19.dat

Output File Name (.EXE)
F:\RBGX_Demo\RRBYW19\RRBYW19.exe

Allow Only One Instance of Compiled Application

Make Evaluation Version
Expiration Date:
6/14/2016

Information Message:

Expiration Message:
The Demo Version has expired.

Update Resources
 External Form Files Plugins Misc. Files

Show Loading Window

Loading
R:BASE X

Load Image Clear Image

Background Color and Font:
 Button Face Font

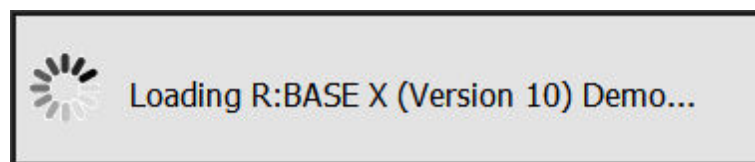
Dimension (W x H):
250 x 75 Show Animation Preview [Esc to Close]

02. Customize the application DAT file to include the following PROPERTY command:

```
PROPERTY LOADWINDOW SHOWINDICATOR 'TRUE'
RETURN
```

03. Compile your application.

04. Once the application is compiled successfully, run the .exe file and notice the Load Window with moving circle.



05. Finally, use the following PROPERTY command as "On After Start" EEP for your main application form to close the loading window.

```
-- On After Start Custom EEP
SET VARIABLE vCompiled TEXT = NULL
GETPROPERTY APPLICATION IsCompiled vCompiled
IF vCompiled = 'YES' THEN
    PROPERTY LOADWINDOW CLOSE 'TRUE'
ENDIF
CLEAR VARIABLE vCompiled
RETURN
```

Have fun implementing new features & enhancements of R:Compiler!

Supported Versions:

- R:BASE X (Build...: 10.0.1.10522 or higher)

16.2 Is the Running Application Compiled or Not?

When you design and then deploy totally automated R:BASE applications, there may be an option for a few users, including yourself, to return to the R:BASE Database Explorer or R> Prompt window. That's fine if your application was started using a full version of R:BASE for Windows.

What happens when you have distributed your cool application using R:Compiler?

You may wish to keep all menu options, and make the optional "Database Explorer, R> Prompt" invisible if the application is using the compiled version of R:BASE. This feature can be achieved without creating separate menus for development and distribution.

Within R:BASE, you may use the "ISCOMPILED" property parameter to capturing whether the application "is compiled" or not. to take advantage the latest versions of R:BASE.

Here's how:

1. Assign a <ComponentID> to a button or menu item of a tree or drop-down menu, such as 'DBExplorer'.
2. Use the following code as "On After Start" Custom EEP.

```
CLEAR VARIABLE vCompiled
SET VARIABLE vCompiled TEXT = NULL
GETPROPERTY APPLICATION ISCOMPILED vCompiled
IF vCompiled = 'YES' THEN
    PROPERTY DBExplorer VISIBLE 'FALSE'
ENDIF
CLEAR VARIABLE vCompiled
RETURN
```

The variable *vCompiled* will return the value of YES or NO.

16.3 Managing Embedded Resources for Compiled Applications

Using the R:Compiler for R:BASE "Additional Resources" tab you may embed the following database and file resources in your compiled executable project:

- External Form Files
- R:BASE Forms
- R:BASE Reports
- R:BASE Labels
- Plugins (.rbm)
- the R:BASE Engine file (.dll)
- Themes (.msstyles, .theme, .uskn, .kskn, .xskn, .mskn)
- R:BASE Command Files (.dat, .rmd, .cmd, .app, .eep, .vie, .frm, .rpt, .lbl, .str, .all, .lst, .tbl)
- R:Charts chart files (.rbc)
- R:BASE Gateway Import/Export Specification files (.rgw)
- R:Mail Editor Template files (.rmt)

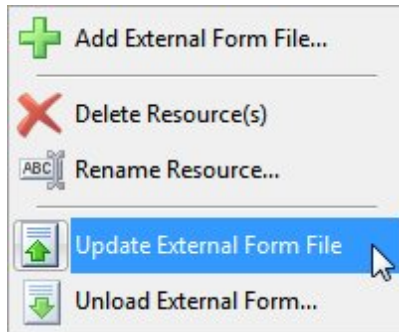
In addition, you may also embed the R:BASE Engine file (.dll) into the compiled executable, as well as define the executable to allow only one instance of the compiled application.

Once the project is deployed, managing embedded resources for your compiled application is as critical as keeping up with the updated versions of R:BASE. In addition to re-compiling and distributing the new compiled executables, you will also need to make sure that you have included the updated versions of all

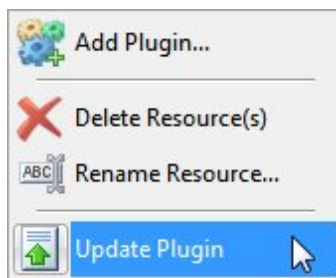
External Form Files, Plugins, and Misc. Files, if updated.

Did you that you can Update External Form Files, Plugins, and Misc. Files by simply using the right-click and select Update option?

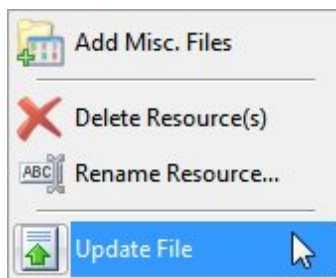
For Updating External Form Files:



For Updating Plugins:



For Updating Misc. Files:



16.4 Reloading Resource Files

File resources can be added to your R:Compiler for R:BASE compiled executable projects, which can include:

- External Forms (.rff)
- R:BASE Plugins (.rbm)
- R:BASE Command Files (.dat, .rmd, .cmd, .app, .eep, .vie, .frm, .rpt, .lbl, .str, .all, .lst, .tbl)
- R:Charts chart files (.rbc)
- R:BASE Gateway Import/Export Specification files (.rgw)
- R:Mail Editor Templates (.rmt)

Did you know that you can simply reload, or update, file resources without having to delete and re-add the resource?

As file resources will change over time, an option to update the file resource is now available. Here's how for Plugins:

01. Open R:Compiler, and select the "Additional Resources" option.
02. Choose the "Plugins" tab, select the appropriate Plugin that needs updated, and click on the "Update Plugin" button.

Here's how for Miscellaneous Files:

01. Open R:Compiler, and select the "Additional Resources" option.
02. Choose the "Misc. Files" tab, select the appropriate file that needs updated, and click on the "Update File" button.

Here's how for External Forms:

01. Open R:Compiler, and select the "Additional Resources" option.
02. Choose the "External Form Files" tab, select the appropriate external form file, and click on the "Update External Form File" button.

As the resource file is listed within R:Compiler, it will assume that the stored location for the file has not changed.

16.5 Saving/Recycling R:Compiler Projects

Did you know that you can now "Save" all settings of your R:Compiler projects, such as, General Properties, EXE-File Icon and Customized Version Information to a file?

R:Compiler includes the features: "Save Project", "Save Project As..." and "Open Project...".

This will help manage different versions of compiled applications and recycle custom details, etc., for your application updates.

16.6 Understanding R:Compiler Resources

When embedding R:BASE Forms, Reports, and Labels as "Additional Resources", the following will assist in your use of R:Compiler for R:BASE:

1. Multiple objects can be selected in the R:BASE Database Explorer to copy into your R:Compiler project. Use the [Shift+Arrow] keys or the [Ctrl] key with the left mouse button to select more than one object.
2. Only [Ctrl+C] at the R:BASE Database Explorer can be used to send your R:BASE resources to the clipboard for R:Compiler to recognize the objects.
3. Only one "type" of object can be added at a time. One or more forms must be added separately than when adding reports or labels.
4. All listed resources can be cleared by using the "Clear Resources" button.
5. After adding Forms/Reports/Labels as resources, make a backup copy of the database, delete those added resources from the database, and reload the database to decrease the overall size and improve network speed.
6. If a resource has been removed from the database, and needs replaced for editing, the item can be copied from the R:Compiler project to the Database Explorer. Use the "Copy to Clipboard" button in R:Compiler, and then [Ctrl+V] at the R:BASE Database Explorer window.
7. When choosing to "Update" an external form file, Plugin, miscellaneous file, R:Compiler will assume that the stored location for the file has not changed.
8. In addition to the Additional Resources toolbar, a speed menu is also available with a right click.

9. When running a compiled executable where a specific resource resides in the executable AND the actual database files, the resource which resides in the executable will be recognized first.

Have fun compiling and deploying R:Compiler applications for R:BASE.

Part



XVII

17 Plugins

Enter topic text here.

17.1 Additional R:Spell Checker Dictionaries on a Network

R:Spell Checker allows for multiple dictionaries to be stored to check spelling. In addition to the English dictionary, Medical, Legal, and Technical dictionaries can be added to verify the accuracy for a word, and to also avoid spell checking commonly terms for medical, legal, and technical business environments.

The Medical terminology dictionary (medical.adm) contains 207,963 words.
 The Legal terminology dictionary (legalwords.adm) contains 4,736 words.
 The Technical terminology dictionary (technical.adm) contains 5,667 words.

The dictionary files would reside into the same folder location as the English dictionary (American.adm).

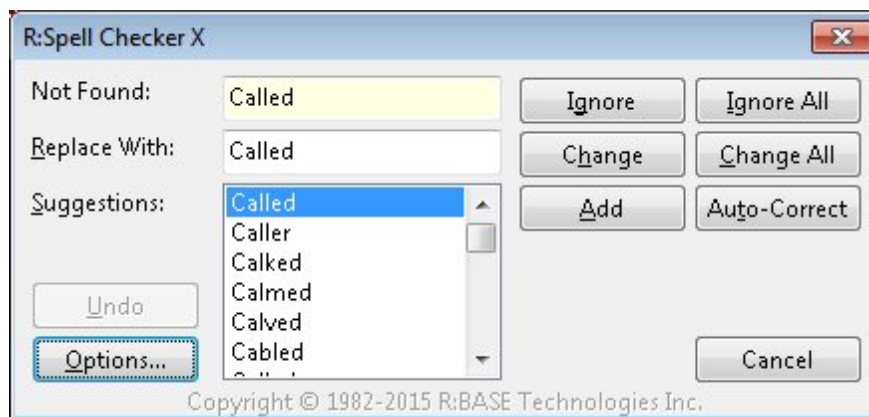
With a multi-user network application and/or compiled application, R:Spell Checker should be configured to use the "CONFIG_TYPE FILE" parameter to perform all spell checking using a RSpellCheckerX.CFG configuration file.

The PLUGIN command syntax would be similar to the following:

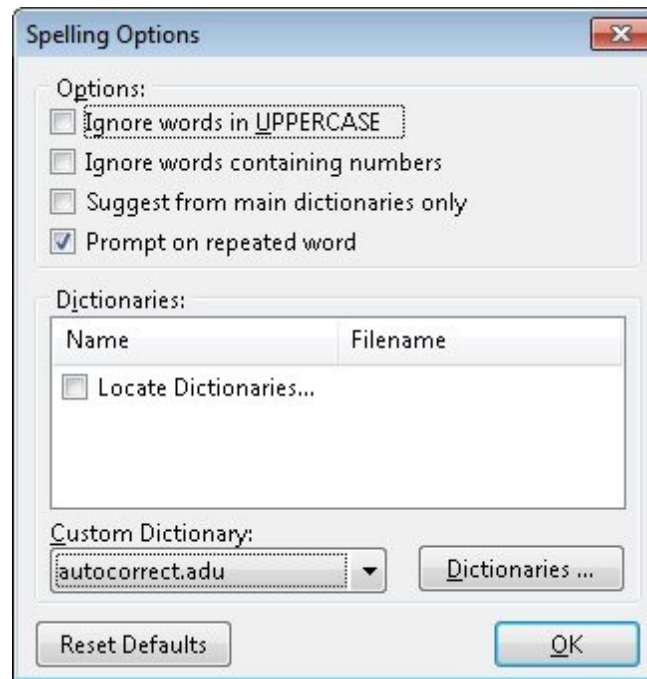
```

    PLUGIN RSpellChecker CallNotes +
    |LANGUAGE ENGLISH +
    |DICTIONARY R:\PHONE_APP\Dictionaries\ +
    |CONFIG_TYPE FILE +
    |CONFIG_FILE R:\PHONE_APP\RSpellCheckerX.CFG
    
```

Using the R:BASE form, perform a spell check upon a misspelled word, to purposely display the R:Spell Checker options.



Select the "Options" button to review the Spelling Options.



In the list of dictionaries, be sure to place a check in the "Medical" or "Technical" dictionary along with "English (US)". If no dictionaries are listed, select the "Locate Dictionaries" option to navigate to the network location where the .adm files are stored. Press OK to save the options.

The new RSpellCheckerX.CFG will then be created, whose contents should be something like the below:

```
[UserName]
_FirstRun=+
_Main_count=2
_Main_0=American.adm
_Custom_count=0
_MSWordCustom_count=0
_ActiveCustom=UserName.adu
_soUppcase=-
_soNumbers=-
_soHTML=-
_soInternet=+
_soQuoted=-
_soAbbreviations=+
_soPrimaryOnly=-
_soRepeated=+
_soLiveSpelling=+
_soLiveCorrect=+
_soUser1=-
_soUser2=-
_soUser3=-
_soUser4=-
_DialogX=926
_DialogY=725
_NewPaths_count=0
_UserData_count=0
```

```
_Main_1=medical.adm
```

Note the two dictionaries, Main_0 and Main_1, are listed.

Any spell checking there after upon medical terms will not be considered misspelled, unless of course if the term itself was entered incorrectly.

17.2 Capturing Common Operating System Folders

Does your application require the ability to send output to a User's directory, only the network contains several different operating systems?

Perhaps the application must add a Favorite to the Internet Explorer menu bar. Or, it is needed that a specific Font be verified?

Access to the all operating system specific common folders like the My Documents, Windows, and System folders are easily accessible using the latest GetCommonFolder Plugin that is part of the R:BASE Plugin Power Pack.

The GetCommonFolder Plugin is used to capture fifty (50) of the most common operating system folders used frequently by applications, but which may not have the same name or location on any given system.

Syntax:

```
PLUGIN GetCommonFolder vFolder|FolderType
```

Where:

vFolder is the variable to return the text value of a given common operating system folder.

FolderType is the name of the specific common folder type.

Examples:

--The following return the Common Programs folder

```
PLUGIN GetCommonFolder vValue|COMMON_PROGRAMS
SET VAR vResult = 'COMMON_PROGRAMS : ' + vValue; WRITE .vResult

COMMON_PROGRAMS : C:\ProgramData\Microsoft\Windows\Start Menu\Programs
```

--The following return the Windows folder

```
PLUGIN GetCommonFolder vValue|WINDOWS
SET VAR vResult = 'WINDOWS : ' + vValue; WRITE .vResult

WINDOWS : C:\Windows
```

--The following return the System folder

```
PLUGIN GetCommonFolder vValue|SYSTEM
SET VAR vResult = 'SYSTEM : ' + vValue; WRITE .vResult

SYSTEM : C:\Windows\system32
```

--The following return the Program Files folder

```
PLUGIN GetCommonFolder vValue|PROGRAM_FILES
SET VAR vResult = 'PROGRAM_FILES : ' + vValue; WRITE .vResult

PROGRAM_FILES : C:\Program Files (x86)
```

In addition to a library of R:BASE Plugins, GetCommonFolder is included with the latest version of the

R:BASE Plugin Power Pack.

R:BASE Plugin Power Pack Online Help Manual:
<http://www.rbase.com/products/PluginPowerPack>

17.3 Converting RTF to Plain TEXT, NOTE or VARCHAR

Did you know that you can convert Rich Text Format (RTF) data to a plain TEXT, NOTE or VARCHAR data using the RTFtoTEXT Plugin?

Syntax: (Four Different Scenarios)

```
-- (01.) When Input/Output Data is a VARIABLE
PLUGIN RRTFtoTXT vResult +
  | INPUT_TYPE VARIABLE +
  | INPUT_VARIABLE inputvarname +
  | OUTPUT_TYPE VARIABLE +
  | OUTPUT_VARIABLE outputvarname +
  | OUTPUT_DATA_TYPE value
```

WHERE:

- vResult is the resulting message value, such as, "OK" or the actual "-ERROR- message"
- INPUT_VARIABLE inputvarname is the input RTF data as variable
- OUTPUT_VARIABLE outputvarname is the resulting output variable name with converted data
- OUTPUT_DATA_TYPE value is the data type (TEXT, NOTE, or VARCHAR)

```
-- (02.) When Input/Output Data is an external FILE
PLUGIN RRTFtoTXT vResult +
  | INPUT_TYPE FILE +
  | INPUT_FILE inputfilename.rtf +
  | OUTPUT_TYPE FILE +
  | OUTPUT_FILE outputfile.txt +
  | OUTPUT_DATA_TYPE value
```

WHERE:

- vResult is the resulting message value, such as, "OK" or the actual "-ERROR- message"
- INPUT_FILE inputfilename.rtf is the input RTF file name
- OUTPUT_FILE outputfilename.txt is the resulting output file with converted data
- OUTPUT_DATA_TYPE value is the data type (TEXT, NOTE, or VARCHAR)

```
-- (03.) When Input is a VARIABLE and Output is an external FILE
PLUGIN RRTFtoTXT vResult +
  | INPUT_TYPE VARIABLE +
  | INPUT_VARIABLE inputvarname +
  | OUTPUT_TYPE FILE +
  | OUTPUT_FILE outputfilename.txt +
  | OUTPUT_DATA_TYPE value
```

WHERE:

- vResult is the resulting message value, such as, "OK" or the actual "-ERROR- message"
- INPUT_VARIABLE inputvarname is the input RTF data as variable
- OUTPUT_FILE outputfilename.txt is the resulting output file with converted data
- OUTPUT_DATA_TYPE value is the data type (TEXT, NOTE, or VARCHAR)

```
-- (04.) When Input is a an external FILE and Output is a VARIABLE
PLUGIN RRTFtoTXT vResult +
  | INPUT_TYPE FILE +
  | INPUT_FILE inputfilename.rtf +
```

```

|OUTPUT_TYPE VARIABLE +
|OUTPUT_VARIABLE outputvarname +
|OUTPUT_DATA_TYPE value

```

WHERE:

- vResult is the resulting message value, such as, "OK" or the actual "-ERROR- message"
- INPUT_FILE inputfilename.rtf is the input RTF file name
- OUTPUT_VARIABLE outputvarname is the resulting output variable name with converted data
- OUTPUT_DATA_TYPE value is the data type (TEXT, NOTE, or VARCHAR)

Examples:

```

-- ConvertingRTF_to_TEXT.RMD
-- Author: A. Razzak Memon
-- Converting_NOTE_VARCHAR.RMD
-- To be used with R:BASE for Windows
-- Required Plugin: RRTFtoTXT
IF (CVAL('DATABASE')) <> 'RTFtoTXT' OR (CVAL('DATABASE')) IS NULL THEN
  CONNECT RTFtoTXT IDENTIFIED BY NONE
ENDIF
LABEL Start
CLS
CLEAR VARIABLES iv%,vRecordID,vRTFData,vTextData,vNoteData, +
  vVarCharData,vResult
SET VARIABLE vRecordID INTEGER = 2
SET VARIABLE vRTFData VARCHAR = NULL
SET VARIABLE vTextData TEXT = NULL
SET VARIABLE vNoteData NOTE = NULL
SET VARIABLE vVarCharData VARCHAR = NULL
SET VARIABLE vResult TEXT = NULL
SELECT RecordID,RTFData INTO +
  vRecordID INDICATOR ivRecordID, +
  vRTFData INDICATOR ivRTFData +
  FROM RTFData WHERE RecordID = .vRecordID

-- Converting RTF Data to TEXT
PLUGIN RRTFtoTXT vResult +
  |INPUT_TYPE VARIABLE +
  |INPUT_VARIABLE vRTFData +
  |OUTPUT_TYPE VARIABLE +
  |OUTPUT_VARIABLE vTextData +
  |OUTPUT_DATA_TYPE TEXT
IF vResult <> 'OK' THEN
  PAUSE 2 USING .vResult ICON ERROR
ELSE
  INSERT INTO TextData VALUES (.vRecordID,.vTextData)
ENDIF

-- Converting RTF Data to NOTE
PLUGIN RRTFtoTXT vResult +
  |INPUT_TYPE VARIABLE +
  |INPUT_VARIABLE vRTFData +
  |OUTPUT_TYPE VARIABLE +
  |OUTPUT_VARIABLE vNoteData +
  |OUTPUT_DATA_TYPE NOTE
IF vResult <> 'OK' THEN

```

```

    PAUSE 2 USING .vResult ICON ERROR
ELSE
    INSERT INTO NoteData VALUES (.vRecordID,.vNoteData)
ENDIF

-- Converting RTF Data to VARCHAR
PLUGIN RRTFtoTXT vResult +
| INPUT_TYPE VARIABLE +
| INPUT_VARIABLE vRTFData +
| OUTPUT_TYPE VARIABLE +
| OUTPUT_VARIABLE vVarCharData +
| OUTPUT_DATA_TYPE VARCHAR
IF vResult <> 'OK' THEN
    PAUSE 2 USING .vResult ICON ERROR
ELSE
    INSERT INTO VarCharData VALUES (.vRecordID,.vVarCharData)
ENDIF
LABEL Done
CLEAR VARIABLES iv%,vRecordID,vRTFData,vTextData,vNoteData, +
    vVarCharData,vResult
RETURN

```

17.4 Creating a Playlist for Windows Media Player

Did you know that you can create a playlist for the Windows Media Player ActiveX Control and then play several files in a continuous mode using the PROPERTY command in form?

Here's how:

1. Using RBEedit or any text editor, create a file with ".m3u" extension, such as:

```

-- PlayList.m3u
D:\MyStuff\Dave_Brubeck_Take5.mp3
D:\MyStuff\Shine on You Crazy Diamond.mp3
D:\MyStuff\Pink_Floyd_Live_At_Pompeii.mpg
D:\MyStuff\Live 8 Performance - Pink Floyd.mpg
D:\MyStuff\R:BASE for Windows 101.avi
D:\MyStuff\R:BASE for Windows Step-by-Step.avi

```

Notice that you can mix and match all sorts of formats supported by Windows Media Player.

2. Save the file as PlayList.m3u
3. Download the sample application posted on 02/14/2006

<http://www.razzak.com/fte>

<http://www.razzak.com/sampleapplications>

"Integrating Windows Media Player ActiveX Control in Forms"

4. Use the following PROPERTY command to dynamically load the playlist.

```

PROPERTY MediaPlayer 'CALL_PROCEDURE CLOSE' ''
PROPERTY MediaPlayer 'URL' PlayList.m3u

```

5. You may also modify the "Load File" routine in sample application to manually load a playlist as follows:


```
-- Load File routine for Windows Media Player ActiveX
CLEAR VAR vFileName
SET VAR vFileName TEXT = NULL
PLUGINS LoadFileName.RBL vFileName +
  |FULLPATH ON +
  |TITLE Select File +
  |NO_LONG_NAMES OFF +
  |FILTER +
  |AVI Files (*.AVI)#*.AVI# +
  |PLAY List Files (*.M3U)#*.M3U# +
  |MP3 Files (*.MP3)#*.MP3# +
  |MPEG Files (*.MPG)#*.MPG# +
  |WMV Files (*.WMV)#*.WMV# +
  |All Files (*.*)#*. * +
  |INITIAL_DIR C:\
IF (vFileName = NULL) OR (vFileName = '[Esc]') THEN
  RETURN
ENDIF
PROPERTY MediaPlayer 'CALL_PROCEDURE CLOSE' ''
PROPERTY MediaPlayer 'URL' .vFileName
RETURN
```

17.5 Dynamically Assigning R:Charts' Tables/Views

By design, each series used in R:Charts is associated with a table or view. The designated table or view is stored with the resulting R:BASE Chart (.RBC) file.

In development, there can be instances where you would like to recycle that cool chart with the exact same column names but with a different table or view, without copying or re-designing a whole new chart. Did you know that you can dynamically change the assigned table or view name for a chart already designed in R:Charts?

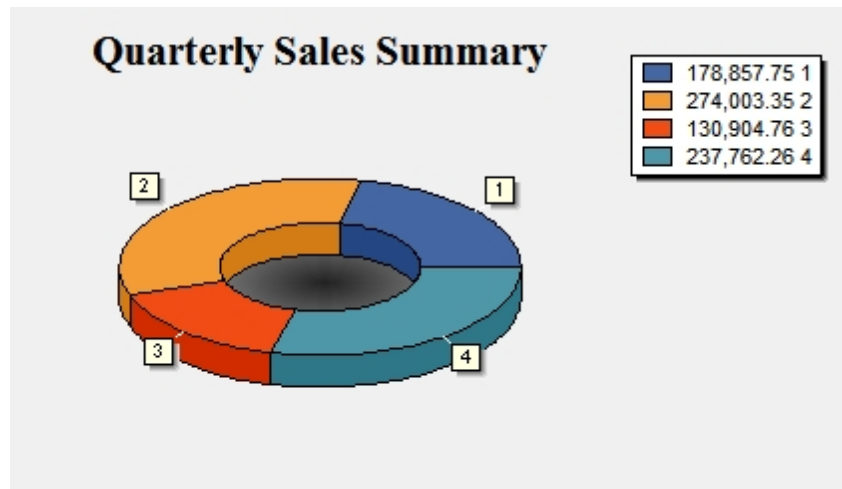
Using the latest R:BASE and R:Charts, now you can use the additional parameter of "SERIES_TABLE[n] tableviewname" to dynamically change the assigned table or view name for any series used in R:Charts, where [n] represents the specific table or view listed in the series.

Example 01:

The R:Chart file (Quarterly_DonutChart.RBC) is based on a QuarterlySummary view in RRBYW19. To print this chart, the following series of commands is used:

```
PLUGIN RCharts +
FILENAME Quarterly_DonutChart.RBC +
|SAVE_TO_JPG Quarterly_DonutChart.jpg +
|IMAGE_WIDTH 424 +
|IMAGE_HEIGHT 244
SET VAR vImage = 'Quarterly_DonutChart.jpg'
RETURN
```

The resulting image is:



Example 02:

To use that exact Quarterly_DonutChart.RBC chart with a different view, for example QuarterlySummary2015, the following series of commands can be used:

```

PLUGIN RCharts +
FILENAME Quarterly_DonutChart.RBC +
|SAVE_TO_JPG Quarterly_DonutChart.jpg +
|IMAGE_WIDTH 424 +
|IMAGE_HEIGHT 244 +
|SERIES_TABLE[0] QuarterlySummary2015
SET VAR vImage = 'Quarterly_DonutChart.jpg'
RETURN

```

17.6 Finding the Row Size using the RRowSize Plugin

Here's how to find Row Size using RRowSize Plugin.

You may integrate this technique in your own database/application.

Example 01:

```

-- Start
IF (CVAL('DATABASE')) <> 'RRBYW19' OR (CVAL('DATABASE')) IS NULL THEN
  CONNECT RRBYW19 IDENTIFIED BY NONE
ENDIF
LABEL Start
CLEAR VAR
vTableName, vRowSize, vString, vPauseMessage, vYesNo, vCaption, vEndKey
SET VAR vTableName TEXT = NULL
SET VAR vRowSize INTEGER = NULL
SET VAR vString TEXT = NULL
SET VAR vPauseMessage TEXT = NULL
SET VAR vYesNo TEXT = 'No'
SET VAR vCaption TEXT = ' Finding Row Size - Select Table Name'
CLS
CHOOSE vTableName FOR #TABLES +
  CHKBOX 1 +
  TITLE 'Select a table and then click on [OK] button to continue' +

```

```

CAPTION .vCaption LINES 19 FORMATTED +
OPTION LIST_FONT_COLOR MAROON +
|TITLE_FONT_COLOR MAROON +
|TITLE_BACK_COLOR [R234,G234,B234] +
|WINDOW_BACK_COLOR [R234,G234,B234] +
|TITLE_FONT_SIZE 10 +
|TITLE_FONT_NAME VERDANA +
|LIST_BACK_COLOR [R234,G234,B234] +
|BUTTONS_SHOW_GLYPH ON +
|THEMENAME R:BASE Rocks!
IF vTableName = '[Esc]' THEN
  GOTO Done
ENDIF
IF vTableName IS NULL THEN
  PAUSE 2 USING 'Table NOT Selected!' +
  CAPTION .vCaption ICON STOP +
  BUTTON 'Click here to select a table ...' +
  OPTION BACK_COLOR WHITE +
  |MESSAGE_COLOR WHITE +
  |MESSAGE_FONT_NAME Verdana +
  |MESSAGE_FONT_COLOR RED +
  |MESSAGE_FONT_SIZE 10 +
  |MESSAGE_FONT_BOLD ON +
  |MESSAGE_FONT_ITALIC OFF +
  |MESSAGE_FONT_STRIKEOUT OFF +
  |BUTTON_COLOR WHITE +
  |BUTTON_FONT_COLOR GREEN +
  |THEMENAME R:BASE Rocks!
  GOTO Start
ENDIF
SET VAR vString = ('vRowSize|TABLE_NAME'&.vTableName)
PLUGIN RRowSize &vString
SET VAR vPauseMessage = +
  ((CHAR(013))+'Table Name:'+(CHAR(009))&.vTableName+(CHAR(013)))+ +
  'Row Size:'+(CHAR(009))&(CTXT(.vRowSize)))
CLS
PAUSE 2 USING .vPauseMessage +
CAPTION 'Finding Row Size' +
ICON APP +
BUTTON 'Press any key to continue ...' +
OPTION MESSAGE_FONT_NAME VERDANA +
|MESSAGE_FONT_SIZE 10 +
|MESSAGE_FONT_COLOR MAROON +
|MESSAGE_FONT_BOLD OFF +
|BUTTON_COLOR WHITE +
|BUTTON_FONT_COLOR GREEN +
|THEMENAME R:BASE Rocks!
GOTO Start
LABEL Done
CLEAR VAR
vTableName,vRowSize,vString,vPauseMessage,vYesNo,vCaption,vEndKey
RETURN
-- End

```

17.7 Sending E-Mail Through MS Outlook With R:BASE MAPI Mail Plugins

When working with the R:BASE MAPI Mail Plugins to send E-Mail using MS Outlook, several additional parameter options are available to ensure the message is sent or is displayed appropriately.

The following are the standard email parameters with the MapiMail3 Plugin:

- TO: - the message recipient(s)
- CC: - the carbon copy recipient(s)
- BCC: - the blind carbon copy recipient(s)
- SUBJECT: - the message subject
- BODY: - the message body content
- ATTACHMENTS: - the file attachment(s)

The following are the email client parameters (MS Outlook) with the MapiMail3 Plugin:

- PROFILE_NAME: - email profile to be used by the Messaging API when the Send method is called. Typically, this parameter is left blank which instructs MAPI to use the default profile.
- PASSWORD: - the email profile password for the login process
- RESOLVE_NAMES: - resolves a message recipient's name (as entered by a user) to an unambiguous address list entry, optionally prompting the user to choose between possible alternative entries, if necessary. (TRUE/FALSE)
- SUPPRESSOUTLOOKSECURITY: - suppresses the MS Outlook security warning (TRUE/FALSE)

The following example demonstrates the use of all parameters with several email recipients:

```
-- MapiMail3_Outlook_With_Attachments.RMD
CLEAR ALL VARIABLES
-- Pre-Define All Variables with Correct Data Type
SET VAR vQuote TEXT = NULL
SET VAR vEMail TEXT = NULL
SET VAR vPipe TEXT = NULL
SET VAR vCRLF TEXT = NULL
SET VAR vTo TEXT = NULL
SET VAR vSubject TEXT = NULL
SET VAR vCc TEXT = NULL
SET VAR vBcc TEXT = NULL
SET VAR vBody TEXT = NULL
SET VAR vAttachments TEXT = NULL
SET VAR vFileSpec TEXT = NULL
SET VAR vToRecipient1 TEXT = NULL
SET VAR vToRecipient2 TEXT = NULL
SET VAR vCCRecipient1 TEXT = NULL
SET VAR vCCRecipient2 TEXT = NULL
SET VAR vBCCRecipient1 TEXT = NULL
SET VAR vBCCRecipient2 TEXT = NULL
SET VAR vToRecipientList TEXT = NULL
SET VAR vCCRecipientList TEXT = NULL
SET VAR vBCCRecipientList TEXT = NULL
SET VAR vResolveNames TEXT = NULL
SET VAR vSuppressSecurity TEXT = NULL
SET VAR vParameters TEXT = NULL
-- Assign Values
SET VAR vQuote = (CVAL('QUOTES'))
```

```

SET VAR vEMail = 'vEMail'
SET VAR vPipe = '|'
SET VAR vCRLF = ((CHAR(13))+(CHAR(10)))
SET VAR vToRecipient1 = 'pterrell@rbtiroasters.com'
SET VAR vToRecipient2 = 'jtaggart@rbtiroasters.com'
SET VAR vToRecipient3 = 'dmachine@rbtiroasters.com'
SET VAR vCCRecipient1 = 'gkennedy@inhouseinspections.com'
SET VAR vCCRecipient2 = 'clidell@rbtiroasters.com'
SET VAR vBCCRecipient1 = 'contracts@inhouseinspections.com'
SET VAR vBCCRecipient2 = 'accountspayable@inhouseinspections.com'
SET VAR vToRecipientList = +
(.vToRecipient1 + vCRLF + .vToRecipient2 + .vCRLF + .vToRecipient3)
SET VAR vCCRecipientList = +
(.vCCRecipient1 + vCRLF + .vCCRecipient2)
SET VAR vBCCRecipientList = +
(.vBCCRecipient1 + vCRLF + .vBCCRecipient2)
SET VAR vFileSpec = +
'C:\Reports\InspectionReport_RBTI_Roasters_22912_20140521.pdf'
SET VAR vTo = ('TO:' + .vToRecipientList)
SET VAR vSubject = ('SUBJECT: Inspection Report Results')
SET VAR vCc = ('CC:' + .vCCRecipientList)
SET VAR vBcc = ('BCC:' + .vBCCRecipientList)
SET VAR vBody = ('BODY: Please review the attached Inspection Report.')
SET VAR vAttachments = ('ATTACHMENTS:' + .vFileSpec)
SET VAR vResolveNames = ('RESOLVE_NAMES:TRUE')
SET VAR vSuppressSecurity = ('SUPPRESSOUTLOOKSECURITY:TRUE')
SET VAR vParameters = +
(.vQuote+.vEMail&.vPipe+.vTo&.vPipe+.vSubject&.vPipe+.vCc&.vPipe+ +
.vBcc&.vPipe+.vBody&+.vPipe+.vAttachments&.vPipe+.vResolveNames& +
.vPipe+.vSuppressSecurity+.vQuote)
PLUGIN MapiMail3 &vParameters
CLEAR VARIABLES vQuote,vEMail,vPipe,vCRLF,vTo,vSubject,vCc,vBcc, +
vBody,vAttachments,vFileSpec,vToRecipient1,vToRecipient2, +
vToRecipient3,vCCRecipient1,vCCRecipient2,vBCCRecipient1, +
vBCCRecipient2,vToRecipientList,vCCRecipientList, +
vBCCRecipientList,vSuppressSecurity,vResolveNames,vParameters
RETURN

```

Now you can automate the tasks of sending email messages to multiple recipients with attachments.

17.8 Using Enhanced Appearance Options In R:Charts

Did you know that you can enhance the resolution of all R:Charts by using the new "Soft Chart" and "Soft Text" settings?

The new appearance settings in R:Charts allow for an enhanced resolution of chart images to display in forms and reports. The settings, "Soft Chart" and "Soft Text" can be applied to the chart object itself, and to the chart text. When using the settings, Soft Text is interpreted only if Soft Chart is ON/checked.

Using the R:Charts Object Inspector toolbar, with "Chart" selected, the "Soft Chart" and "Soft Text" options are available.

The updated PLUGIN command syntax for R:Charts includes the two additional parameters to enhance the appearance for charts.

SOFT_CHART - specifies to use an enhanced/smooth display for the chart objects (ON/OFF)

SOFT_TEXT - specifies to use an enhanced/smooth display for the chart text (ON/OFF)

```
-- Example:
PLUGIN RCharts vResult+
|FILENAME Quarterly_BarHorizontal.rbc +
|SAVE_TO_JPG Quaterly_Bar.jpg +
|CHART_TITLE Quarterly Summary +
|LEGEND_TITLE Summary +
|SOFT_CHART ON +
|SOFT_TEXT ON
RETURN
```

Supported Versions:

- R:BASE X (Build..: 10.0.1.10627 or higher)
- R:Charts X (Build..: 10.0.1.10627 or higher)

17.9 Using the Enhanced MAPIMail Plugins with Thunderbird

There are several steps required to use the enhanced R:BASE MAPIMail Plugins with Mozilla Thunderbird:

First, you must alter your Windows operating system setting for your default email program as follows:

1. Select the "Start" button, and choose "Settings" > "Control Panel" > "Add/Remove Programs"
2. Click on "Set Program Access and Defaults" button on the left panel.
3. Select the "Custom" radio button.
4. Click on the "down arrow" characters to expand the available options.
5. Select the "Mozilla Thunderbird" radio button under "Choose a default e-mail program".
6. Click on the "OK" button to save your changes.

Next, you must update the Windows Registry in order to make sure that the MAPI protocol is installed:

1. Select the "Start" button for your Windows operating system, and choose "Run".
2. Enter "REGEDIT" into the field, and the Registry Editor will launch.
3. From the list on the left panel, select the value: HKEY_LOCAL_MACHINE.
4. Within the tree view, select sub categories "SOFTWARE" > "Microsoft" > "Windows Messaging SubSystem".
5. On the right panel, verify that the value for "MAPI" does exist and is equal to 1.
6. Close the Registry Editor window.

Now, we will check certain MAPI options within Mozilla Thunderbird

1. Start the Mozilla Thunderbird email client software
2. From the main Menu Bar, select "Tools" > "Options"
3. Within this dialog window, select the "Advanced" button
4. Under the "General" tab, select the "Config Editor..." button
5. Type "mapi" into the filter to see the parameters that contain the word mapi.
6. Verify that the following parameters exist:

```
mapi.blind-send.enabled = true
mapi.blind-send.warn = false
```

The first parameter, *mapi.blind-send.enabled*, will likely already exist with its value set to true. The second parameter may need to be added.

1. If either parameter needs added, right click in the white space and select "New" > " Boolean".
2. Enter the parameter, i.e. "mapi.blind-send.warn", and select the "OK" button.
3. From the true/false options, choose the appropriate value based upon what is listed above for that parameter.
4. In order to verify your new parameter, click the "Show All" button, and reenter the "mapi"

filter.

Finally, you use the Enhanced MAPIMail Plugins (included with Update 1 of the R:BASE Plugin Power Pack).

1. Copy the following MAPIMail Plugins into the appropriate R:BASE program directory or compiled application directory. Here are the file names and the differences between the Plugins.
 - MAPIMail.RBL (interactive - appears under Utilities > Plugins within R:BASE)
 - MapiMail2.RBL (automated - opens the MAPIMail dialog window and displays the defined parameters)
 - MapiMail3.RBL (automated - opens the default email client and displays the defined parameters)
 - MapiMail4.RBL (automated - sends the email message automatically)
2. Using the automated Plugin parameters

Syntax:

```
PLUGINS MapiMail2.rbl vMail|<parameters>
PLUGINS MapiMail3.rbl vMail|<parameters>
PLUGINS MapiMail4.rbl vMail|<parameters>
```

Parameters:

TO:	specifies the list of recipients of the message
CC:	specifies the list of carbon copy recipients of the message
BCC:	specifies the list of blind carbon copy recipients of the message
SUBJECT:	specifies the subject string for the message
BODY:	specifies the body content of the message
ATTACHMENTS:	specifies the list of files to be send along with the message
PROFILE_NAME:	specifies a Profile to be used by the Messaging API when the Send method is called. Typically, this parameter is left blank which instructs MAPI to use the default profile.
PASSWORD:	specifies the password to be used during the logon process
SENDDIRECT:	specifies the setting for the "Send Directly" check box (TRUE/FALSE)
RESOLVE_NAMES:	specifies the setting to resolve a message recipient's name (as entered by a user) to an unambiguous address list entry, optionally prompting the user to choose between possible entries, if necessary. (TRUE/FALSE)

Where:

vMail is the text variable to return the status, such as 'OK' or the exact -ERROR- message

Notes:

Returned variable name and the exact file name with path must be separated by a "|" pipe symbol.

17.10 Using the R:AudioPlayer Plugin

The R:AudioPlayer Plugin is used to play audio files. The Plugin supports the following audio formats:

- AVI
- MDI
- MP3
- WAV
- WMA

Syntax:

```
PLUGIN RAudioPlayer vResult|option
```

Supported Options:

- FILENAME filespec
- PLAY
- PAUSE
- RESUME
- STOP

Where:

- vResult is the text variable to return the status, such as 'OK' or the exact -ERROR- message
- FILENAME is the name and path of the audio file
- PLAY is used to play the current file
- PAUSE is used to pause the current file
- RESUME is used to resume playing the current file
- STOP is used to stop playing the current file

Each option must be separated by a "|" pipe symbol.

```
-- Example 01: To load and play audio file
PLUGIN RAudioPlayer vResult|FILENAME DreamTheater_Octavarium.mp3
PLUGIN RAudioPlayer vResult|PLAY
RETURN

-- Example 02: To pause the already playing audio
PLUGIN RAudioPlayer vResult|PAUSE
RETURN

-- Example 03: To resume the paused audio
PLUGIN RAudioPlayer vResult|RESUME
RETURN

-- Example 04: To stop the audio player
PLUGIN RAudioPlayer vResult|STOP
RETURN
```

In order to use R:Audio Player Plugin with R:BASE, you have to make sure that the version specific Plugin is installed in the R:BASE program folder, or in the compiled (.exe) application folder. R:AudioPlayer does not appear in the R:BASE for Windows Plugins Menu.

Some Plugins are only made available through the purchase of the R:BASE Plugin Power Pack.
<http://www.rbase.com/products/PluginPowerPack>

Part

XVIII

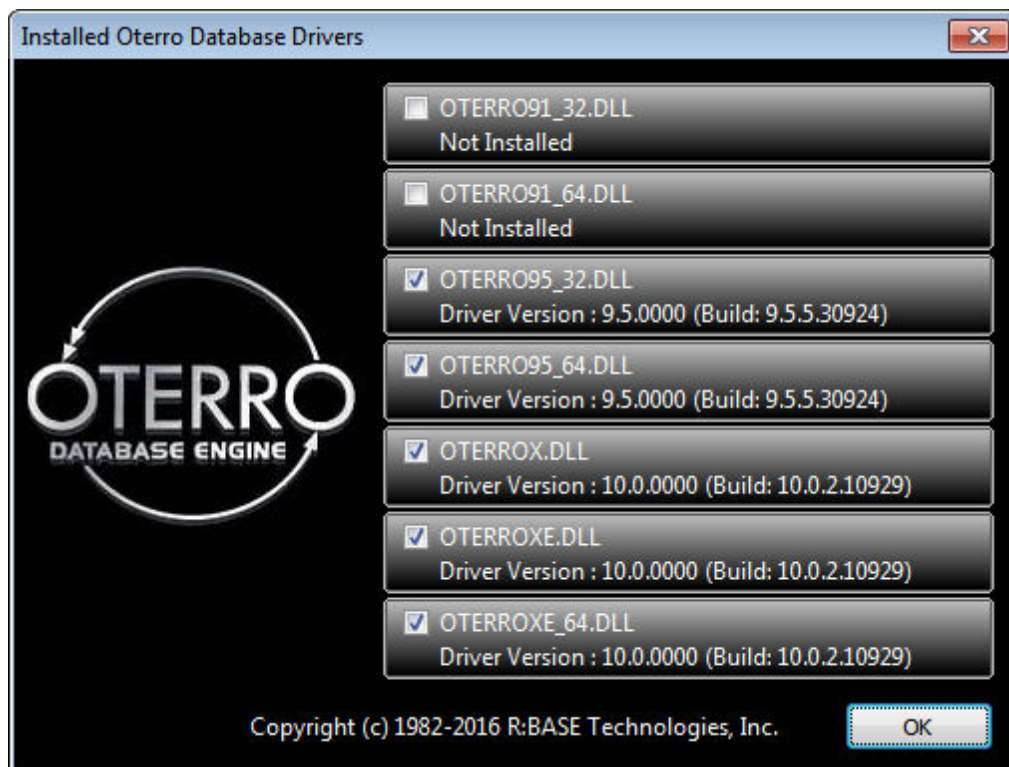
18 Oterro / ODBC

18.1 Checking Version and Build of Oterro

With various versions of Oterro installed on your Network Server, Workstation, Notebook, or Tablet PC, it may be helpful to know the exact version and build of installed version(s) of Oterro.

Did you know that you use the "CheckVer" utility installed in the default installation folder of Oterro?

After installing/updating the latest version of Oterro, you will find the enhanced "CheckVer" utility that will illustrate all installed versions of Oterro ODBC drivers, including the file name, driver version and build of Oterro.



18.2 Configuring Oterro and R:BASE ODBC Drivers on 64-bit OS

If you are running any of the R:BASE products above on MS XP (64), Vista (64), Win 7 Professional (64), or Windows Server 2003/2008 (64), you may need to tweak the registry settings to recognize installed R:BASE and Oterro ODBC settings and installed drivers.

Typically, using the full installer, after a successful installation of R:BASE/Oterro, you would expect the R:BASE/Oterro ODBC drives to be listed under the Drivers for ODBC Data Source Administrator panel. Well, by default, on all XP (64), Vista (64), Windows Server 2003 (64), and Windows Server 2008 R2 x 64 with Service Pack 2 machines, the installed ODBC Drivers list is empty. There are no drivers listed, even for Microsoft's own default drivers.

The reason the installed drivers are not listed is the way Windowsx64 editions operate with the ODBC drivers. On Windows x64 editions, all ODBC drivers are installed in the "%SystemRoot%\SysWOW64" folder.

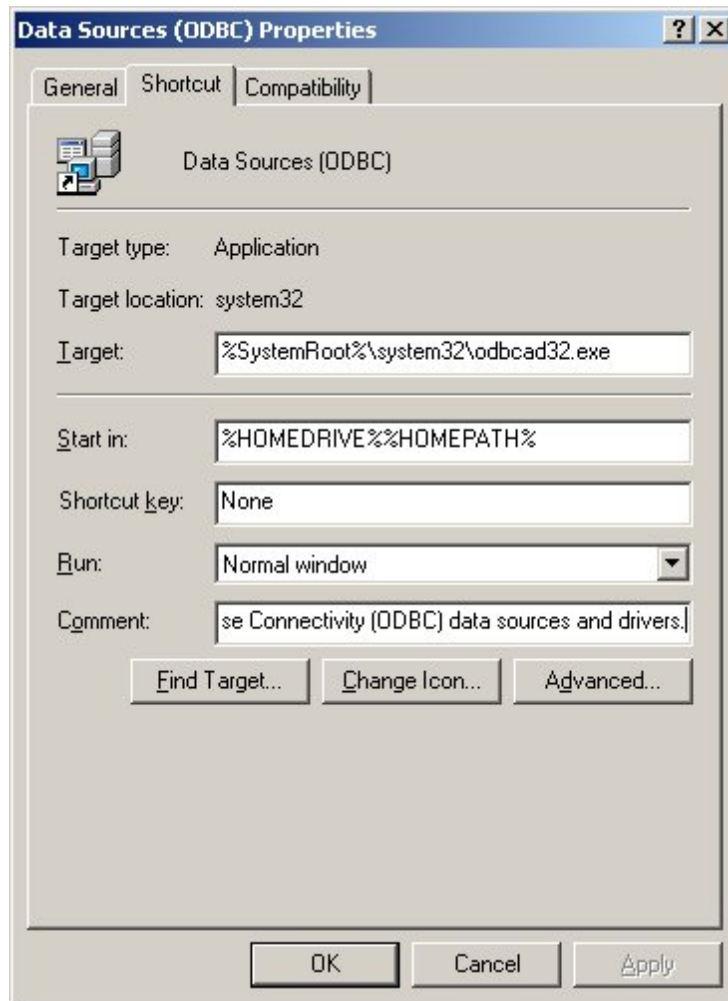
However, the default ODBC Data Source Administrator that is available through the Administrative Tools

or Control Panel in Windows x64 Editions is located in the "%SystemRoot%\system32" folder, and only searches this folder for ODBC drivers.

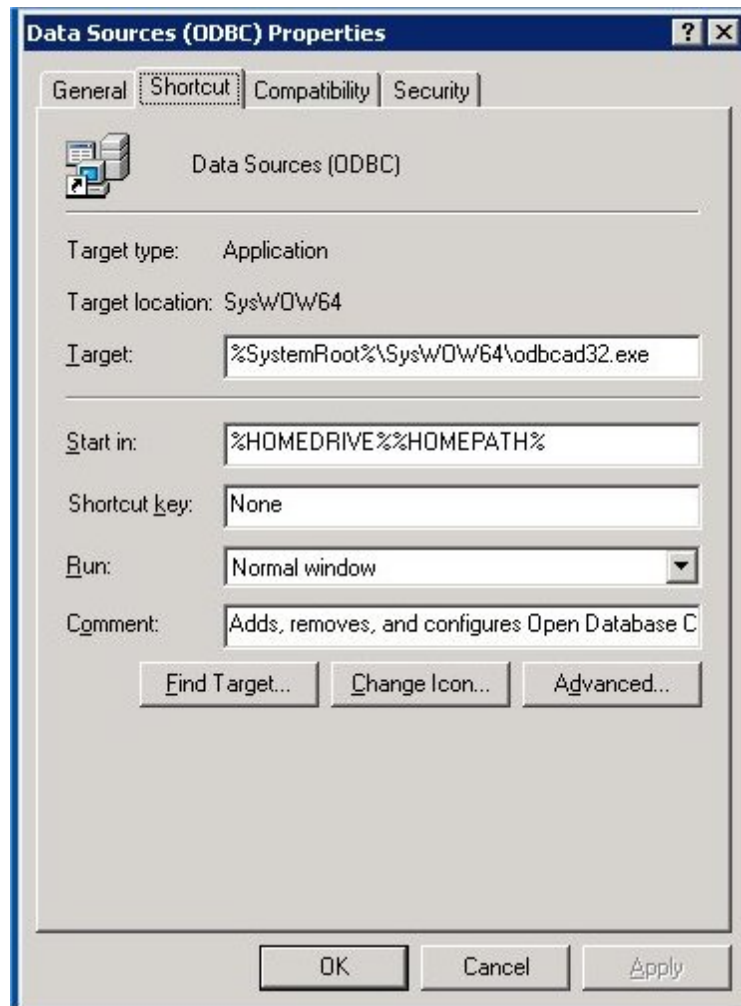
To alter this behavior on Windows x64 editions, you should update the target information for shortcut properties of the ODBC Data Source Administrator tool to reflect "%SystemRoot%\SysWOW64\odbcad32.exe".

This will correctly locate and list all installed ODBC drivers as well as enable you to create a System DSN or should be recognized when using the DSN-Less connection.

Default Properties for ODBC Data Source Administrator



Modified Properties for ODBC Data Source Administrator



Have fun running R:BASE and Oterro on Windows XP (64), Vista (64), Win 7 Professional (64), Win Server 2003 R2 x64, and Win Server 2008 (64) operating systems!

18.3 Using UPDATE Triggers with Oterro

R:BASE Trigger is procedural code that is automatically executed in response to certain events on a particular table in the database.

Triggers can restrict access to specific data, perform logging, or audit data modifications.

There are "BEFORE" triggers and "AFTER" triggers which identifies the time of execution of the trigger. A trigger can be set to automatically run a Stored Procedure BEFORE and/or AFTER an UPDATE, DELETE, or INSERT event occurs in a table.

There are three triggering events that cause triggers to fire:

- INSERT event (as a new record is being inserted into the database)
- UPDATE event (as a record is being changed)
- DELETE event (as a record is being deleted)

You may define specific Triggers using the Triggers section of the Data Designer:

Database Explorer | Tables | Design Table | Table Designer | Triggers

- Before Insert
- After Insert
- Before Update
- After Update
- Before Delete
- After Delete

When designing web applications for your R:BASE database, using various web development tools, a direct connection to R:BASE database is established via System DSN using Oterro.

Yes, now you can use your favorite web development tool that will properly execute all Triggers, if defined, including the UPDATE trigger.

18.4 Which ODBC Driver to Use

Each licensed version of R:BASE typical installation also includes the native R:BASE ODBC driver to be used with the licensed version of R:BASE ONLY and not with any other apps such as R:Compiled or Runtime applications, etc.

To use an ODBC driver for your Compiled or Runtime applications, you should purchase a separate license of Oterro accordingly. Using the R:BASE ODBC Driver with your R:Compiled or Runtime applications would be a clear violation of the R:BASE license.

Here are the differences (other than number of connections) between the Oterro and R:BASE ODBC drivers:

The R:BASE ODBC driver does not support updates, deletes, or inserts done with a positioned cursor, while Oterro does.

The R:BASE ODBC driver only supports the following commands through SQLExec:

CLEAR:
DECLARE:
DELETE:
INSERT:
SELECT:
SET:
UPDATE:

The Oterro driver supports the above commands plus the following:

ALTER:
APPEND:
ATTACH:
AUTOCHK:
AUTONUM:
CALL:
COMMENT:
CREATE:
DETACH:
DROP:
GET:
GRANT:
IF:
INPUT:
LAUNCH:
LOAD:
OUTPUT:
PACK:
PROJECT:
PUT:
RENAME:
REVOKE:
REVOKE:

RULES:
SATTACH:
SCONNECT:
SDETACH:
SDISCONNECT:
UNLOAD:
WHILE:

Based upon your needs, now you many use the appropriate ODBC driver as you see fit.

Part

XIX

19 Feedback

Suggestions and Enhancement Requests:

From time to time, everyone comes up with an idea for something they'd like their software to do differently.

If you come across an idea that you think might make a nice enhancement, your input is always welcome.

Please submit your suggestion and/or enhancement request to the R:BASE Developers' Corner Crew (R:DCC) and describe what you think might make a nice enhancement. In R:BASE, the R:DCC Client is fully integrated to communicate with the R:BASE development team. From the main Menu Bar, choose "Help" > "R:DCC Client". If you do not have a login profile, select "New User" to create one.

If you have a sample you wish to provide, have the files prepared within a zip archive prior to initiating the request. You will be prompted to upload any attachments during the submission process.

Unless additional information is needed, you will not receive a direct response. You can periodically check the status of your submitted enhancement request.

If you are experiencing any difficulties with the R:DCC Client, please send an e-mail to rdcc@rbase.com.

Reporting Bugs:

If you experience something you think might be a bug, please report it to the R:BASE Developers' Corner Crew. In R:BASE, the R:DCC Client is fully integrated to communicate with the R:BASE development team. From the main Menu Bar, choose "Help" > "R:DCC Client". If you do not have a login profile, select "New User" to create one.

You will need to describe:

- What you did, what happened, and what you expected to happen
- The product version and build
- Any error messages displayed
- What computer operating system is in use
- Anything else you think might be relevant

If you have a sample you wish to provide, have the files prepared within a zip archive prior to initiating the bug report. You will be prompted to upload any attachments during the submission process.

Unless additional information is needed, you will not receive a direct response. You can periodically check the status of your submitted bug.

If you are experiencing any difficulties with the R:DCC Client, please send an e-mail to rdcc@rbase.com.

Index

- # -

#DATE 24
 #NOW 7, 24, 40, 250
 #PI 24
 #TIME 24
 #WHERE
 theme 230

- . -

.CFG 14
 .msstyles 18

- A -

accelerator character 157
 Action
 On Before Design 54
 actions
 After Generate 163
 Before Generate 163
 On Before Design 119
 ActiveX 137, 267
 ADDMON 234
 Advanced Variable Rich Edit 95
 ALTER TABLE
 CASCADE 194
 ampersand 157
 ANSI 184
 Appearance 64, 113
 application
 compiled 16, 257
 hide 207
 Runtime 16
 Application File
 convert 103
 ARRANGE BY 167
 ATTACH
 column alias 226
 table alias 226
 temporary 226
 Audio Player 274

- B -

background color 144
 backup 22
 BiDiMode 149
 BIGINT 19
 binary 210
 Bit Button
 load image 134
 Blank If Zero 89
 BLOB Editor 18
 caption 189
 default tab 225
 icon 189
 BOOLEAN 19
 BROWSE 212, 215
 EXPORTRESULT 83
 NOCHG 83
 PRINTDATA 83
 READONLY 83
 SAVERESULT 83
 buttons 197

- C -

calendar 148
 Change Tab Order 147
 character
 continuation 61
 character case
 lower case 110
 sentence case 110
 title case 110
 upper case 110
 Check Box
 bidirectional mode 149
 default state 149
 CHECKPROP 57
 CHKCUR 246
 CHKFUNC 238
 CHKVAR 246
 CHOOSE 210, 219
 CLEAR
 TABLE LOCKS 195
 clipboard 61, 193, 210
 copy 107

clipboard 61, 193, 210
 cut 107
 paste 107
 CLS 61
 code
 comment 180
 uncomment 180
 COLCHECK 45
 color 141
 custom 17
 extended 35
 integer value 6, 35
 palette 34
 RGB value 6, 35
 column name 98
 Combo Box 100
 command 219
 Command History 32
 load 61
 save 61
 Comment 144
 Component ID 9, 99
 default 45
 computed column 2
 Computer 16
 configuration file 14, 46
 CONNECT 14
 Consolidated Control Palette 165
 control code 170
 Control Palette 165
 Control Type 13
 Controls 100
 conversion
 data types 265
 CREATE TABLE 80
 CREATEOBJECT 13, 108
 CrossTab
 color 160
 titles 162
 cursor focus 144
 Custom EEP 9, 138
 Custom EEPs 118
 Custom Form Action 97, 103
 copy 105
 paste 105
 CVAL 10, 16, 57, 210, 234
 DATABASE 11
 EDITOR 184

LINES 192
 NETGROUP 237
 NETLOCALGROUP 237
 NETUSER 11
 PORTS 235
 ROWCOUNT 192
 SCRATCH 53
 SHORTNAM 226
 VERSION 23
 CVTYPE 236

- D -

Data Browser/Editor 69, 83
 average 89
 calculate 89
 compute 89
 count 89
 duplicate row 83
 duplicates 89
 maximum 89
 minimum 89
 standard deviation 89
 sum 89
 tally 89
 variance 89
 Data Designer
 computed column 80
 default value 80
 reorder column 80
 Data Dictionary 2, 6, 8, 13, 18, 34, 35, 133
 Database Settings 187
 data type 19
 database
 maintenance 14
 Database Explorer 64, 75
 find 66, 68
 rename object 70
 search 66, 68, 74
 Database Explorer search
 comments 68
 control properties 68
 Custom EEP 68
 expressions 68
 name 68
 database files 3
 date 98
 DATETIME 98

DB Date/Time Picker 102
DB Edit 98, 110
 balloon tip 121
 cue banner 121
 hint 121
 watermark 121
DB Email Label
 BCC 95
 CC 95
 Message Body 95
DB Grid 103, 110
 column properties 110
 date picker 148
DB Image 153
DB Lookup List View 115
 highlight 115
 selected row colors 115
DB Spinner 154
DB Tree View 143
DCX 18
default
 Open Image dialog 7
 paper size 47
 paper source 47
 printer 47
default setting 149
DELETEOBJECT 108
DELFUNC 238
deploy application 7
design password 169
desktop shortcut
 properties 38
 Start in 38
 Target 38
DIALOG 193, 197
 calculator 216
 hints 204
 input color 218
 lookup 204
 popup 198, 216
 theme 204, 216
dictionary
 Legal 262
 Medical 262
 Technical 262
Digital Clock 134
DIR 226
Directory Tree 156

display format 170
DLCALL 238
DLL
 version 42
DROP
 NOCHECK 203
 VIEW 203
DROP ALL 218
Drop-Down Button
 load image 134
dual-monitor 208

- E -

EDIT 215
EEP 138, 152
 action sequence 95
 document 9, 125, 164
 On Decrement 154
 On Increment 154
 On Key Press 157
 On Mouse 153
 On Right Click 154
email 273
e-mail 273
embedded
 LOADWINDOW 255
 SHOWINDICATOR 255
encoding 184
ENCRYPT 196
encryption 196, 203
Engine
 version 42
Enhanced DB Grid 98, 103, 141, 152
 band list 126
 customize filters 131
 disable column sort 120
 Filter Bar 111, 128
 focus on Filter Bar 132
 hot keys 131
 multi-column sort 151
 record number 124
 search records 131
 show/hide columns 131
 sort column 152
Enhanced Group Box
 transparent 134
error checking 57

exceptions 3
 executable 3
 parameters 38
 export
 footer text 174
 header text 174
 multi-tab 175
 specification file 177
 workbook 175
 worksheet 174
 Expression Builder 8
 Calculate On 161
 copy 105, 161
 paste 105, 161
 external file 210
 External Form 217
 External Form File 103, 105, 144
 comment 70
 version 70

- F -

FASTLOCK 20
 feedback 283
 field color 141
 file
 scratch 46
 temporary 46, 53
 file encoding 184
 File Gateway
 specifications 177
 File List Box 124, 150, 155
 file mask 150
 Files In Use 42
 filter 152
 FILTERTYPE 152
 find 143
 control properties 71
 expression 73
 Find in Files 41, 77
 FINDNODE 143
 foreign key 8, 36, 208
 form
 closing behavior 112
 copy 4
 date/time 29
 drag and drop 130
 preview 68

 scaling 155
 table 136
 unload 65
 variable 8
 version 29
 form controls
 Component ID 133
 convert 103
 Form Designer 107
 display format 153
 form EEP
 document 125
 Form Properties
 EEPs 118
 Roll Up Button 118
 Form Wizard 149
 FORMAT 109, 142, 250
 Format Mask 153
 formula 2
 function 242
 function key 97

- G -

GATEWAY 174, 177
 EXPORT 174, 175
 IMPORT 175, 177
 GETDATE 248
 GetIPAddress 16
 GetMACAddress 16
 GETPROPERTY 16, 45, 57, 98, 101, 126, 128,
 132, 137, 155, 156, 160, 163, 165, 166, 172, 208, 257
 Internet connection 232
 GETROPERTY 161
 GETVAL 16
 GetLock 236
 GIF 18
 global groups 237
 GUID 11

- H -

highlight 141
 hint
 balloon 47
 icon 47
 pause 47

hot key 61, 97, 157

- I -

ICAP2 241
 icon 193
 ICON_FILE 193
 IFF 239
 IFGE 241
 IFLE 241
 IFNE 241
 IFWINDOW 242
 IINFO 242
 image 124, 130, 153, 193
 multipage 18
 IMON 234
 import
 column count 174
 column mapping 175
 mapping 175
 specification file 177
 spreadsheet 175
 suppress message 177
 XLS 175
 XLSX 175
 indexe 8
 initial position 219
 initial value 219
 INITPOS 219
 INITVAL 219
 INPUT_LANGUAGE 4
 INT 192
 INTERSECT 207
 ISTR 237
 Item Height 124
 ITEMCNT 210
 IYR4 234

- J -

JOIN 207

- K -

key 97
 key combination 97
 keyboard 4

- L -

label
 copy 4
 date/time 29
 persistent space 171
 unload 65
 variable 8
 version 29
 Label Designer
 toolbar 165
 language 4
 LAUNCH 207, 228
 LIST
 CASCADE 208
 FKEYS 208
 FORMS 206
 FUNCTIONS 238
 LABELS 206
 PKEYS 208
 REPORTS 206
 TABLE 206
 TABLES 202
 UKEYS 208
 List Box 100
 list view 100, 115, 123
 LISTOF 251
 LJS 80
 LOAD 55
 local groups 237
 lookup variable 138

- M -

machine code 170
 Magnifying Glass 41
 MAPI 95, 273
 MAPIMail 271, 273
 mask 150
 MDI 69, 75, 217
 Media Player 137, 267
 Memo Editor 165
 Memo Object 165
 Menu Bar 103, 113
 copy 106
 paste 106

Menu Bar Designer 113
 milliseconds 14
 monitor 208
 mouse over 153
 multi-monitor 208
 multi-user 54
 optimize 20
 performance 56
 settings 45
 table locks 195
 warning 58

- N -

native 144
 NATIVE_FORMS 144
 NETGROUP 16
 NETLOCALGROUP 16
 NetUser 16
 network access 237
 NEXT 45
 NOCALC 55
 NOTE 18

- O -

ODBC 192, 206
 ODBC driver 277
 check version 277
 ON CONNECT 223
 On-Screen Keyboard 4, 5
 Open After Export 89
 operating system
 common folders 264
 optimize 138, 215
 Oterro 277
 OUTPUT 14, 196
 ENCRYPT 203
 PDF 192
 PRINTER 201

- P -

Page Footer 171
 Page Header 171
 page lock 54
 PAGELOCK 54

password 169, 196
 PAUSE 163, 197
 GAUGE 227
 PCC Label 170
 PDF 166
 PDF Viewer 101
 performance 138
 PLUGIN 11, 262, 264, 265, 267, 269, 271, 272,
 273, 274
 Plugin Power Pack 264, 265, 269, 273, 274
 Plugins 271
 pop-up menu
 custom 120
 items 120
 primary key 8, 208
 PRINT
 margins 194
 sorting 167
 theme 163
 watermark 166
 zoom percent 230
 print password 169
 printer 210
 printer control code 170
 printer settings 234
 Printer Setup 166
 printer tray 211
 PRNSETUP 210, 224, 234
 LEFT 229
 standard print window 228
 theme 200
 TOP 229
 tray 211
 program
 hidden 207
 maximized 207
 minimized 207
 normal 207
 PROPERTY 4, 5, 10, 16, 18, 27, 45, 49, 57, 95,
 101, 102, 107, 108, 112, 121, 126, 128, 130, 132, 133,
 134, 136, 137, 143, 144, 147, 148, 150, 152, 155, 160,
 168, 172, 193, 209, 210, 229, 255, 267
 disable add rows 102
 disable delete rows 102
 TABLE 102
 wild card 232

- Q -

QUALCOLS 20, 54
 QUIT TO 209

- R -

R:Backup 22
 R:BASE Application 228
 scale 229
 R:BASE Editor 184
 right margin 185
 run code 184
 Structure Toolbar 181
 wrap commands 185
 R:BASE On-Screen Keyboard 27
 R:Charts 272
 assign table 268
 assign view 268
 R:Compiler project
 open 259
 save 259
 R:Compiler resource
 reload 258
 R:Compiler resources
 embedded 257
 tips 259
 update 257
 R:Spell Checker
 configuration file 262
 R> Prompt 32, 61
 Radio Button
 alignment 97
 bidirectional mode 149
 button count 132
 Radio Group
 alignment 97
 bidirectional mode 149
 Radio Group Box
 spacing 148
 RBA 10, 103
 RBBE 18
 RBBEDIT 195, 225
 RBTI 102
 RBTI_CEM_FONTCOLOR 24
 RBTI_CEM_FONTSIZE 24

RBTI_DBGRID_COLUMN 24
 RBTI_DIRTY_FLAG 24
 RBTI_ElapsedTime 24
 RBTI_ERR_FONTCOLOR 24
 RBTI_ERR_FONTSIZE 24
 RBTI_FORM_ALIAS 24
 RBTI_FORM_CLICKED 24, 102
 RBTI_FORM_COLNAME 24
 RBTI_FORM_COLVALUE 24
 RBTI_FORM_COMPID 24
 RBTI_FORM_CTRLTEXT 24, 102
 RBTI_FORM_DATATYPE 24
 RBTI_FORM_DBLCLKED 24, 102
 RBTI_FORM_DIRTYVAR 24
 RBTI_FORM_DSGN 24
 RBTI_FORM_EXT_DSGN 24
 RBTI_FORM_FORMNAME 24
 RBTI_FORM_MODE 24
 RBTI_FORM_TBLNAME 24
 RBTI_FORM_VARNAME 24
 RBTI_FORM_VARVALUE 24
 RBTI_IMG_FOLDER 7, 24
 RBTI_LABEL_NAME 24
 RBTI_PRNDEFDIR 24
 RBTI_PRNDEFFILE 24
 RBTI_PRNDEFFILET 24
 RBTI_REPORT_NAME 24
 RBTI_RowsDeleted 24
 RBTI_RowsInserted 24
 RBTI_RowsUpdated 24
 RDATE 234
 RECALC 209
 RECALC VARIABLES 209
 recalculate 209
 referential integrity 36
 REFF 217
 regular expression 142
 relative word position 235
 report
 Actions 163, 172
 arrange tables 167
 copy 4
 date/time 29
 EEP 172
 number of pages 165
 page number 161
 persistent space 171
 properties 166

- report
 - unload 65
 - variable 8
 - version 29
 - Report Designer
 - toolbar 165
 - RESIZE_ICON 193
 - resolution 272
 - RFF 103
 - RGUID 11
 - rich text 166, 195
 - RNDDOWN 238
 - Roll Up Form 118
 - ROSK 4, 5, 27
 - row lock 54
 - row number 123
 - row select 141
 - row size 269
 - ROWLOCKS 54
 - RULES 80
 - RUN 14, 209
 - RUNRBA 228
 - Runtime 16
 - RWP 235
- S -**
- SATTACH 192
 - save 210
 - save image 130
 - SCALEBY 155
 - scientific notation 170
 - scratch 3
 - screen shot 193
 - screen size 208
 - script error 147
 - search 143
 - control properties 71
 - expression 73
 - security 237
 - SELECT 138, 192, 201
 - select all 143
 - server table 192
 - SET 57
 - ECHO 51
 - FEEDBACK 230
 - LINES 201, 203
 - NAMEWIDTH 203
 - NULL 201
 - PROGRESS 231
 - SCRATCH 46, 53
 - TRACE 51, 56
 - UINOTIF 56
 - WIDTH 201, 203
 - setting 55, 57, 89
 - close R:BASE warning 58
 - hint 47
 - scratch 46
 - table 102
 - WALKMENU 49
 - settings
 - Character 50
 - CURRENCY 50
 - Data Designer 58
 - database 50
 - Database Events 50
 - Database Explorer 57, 69
 - DATE 50
 - Form Designer 68
 - Operating Conditions 50
 - registry settings 52
 - scratch 209
 - TIME 50
 - Shape Object
 - pen style 168
 - shape style 168
 - SHOW 57
 - SLEN 80
 - SLOC 80
 - SLOCI 247
 - SLOCP 247
 - SMALLINT 19
 - SMOVE 80
 - Soft Chart 272
 - Soft Text 272
 - sort order 98
 - sorting 151
 - Speed Button
 - down status 133
 - group 133
 - SQLCODE 24
 - SQLSTATE 24
 - Startup 38
 - STATICDB 20
 - Status Bar
 - column 124

Status Bar
 status 124
 table 124
 stored procedures 91
 STRIM 80
 Sub-Report 166, 167
 SUBTRACT 207
 SYS_FORMS3 29
 SYS_LABELS3 29
 SYS_REPORTS3 29
 system variable 7, 24, 40
 form 99, 135
 report 160
 system variables 102

- T -

Tab Order 147
 table
 dBASE 206
 lock 206
 server 206
 temporary 119, 206
 table locks 195
 task bar 208
 TEMP 46
 temporary 3
 temporary files 209
 temporary table 36, 192, 207
 temporary tables 202
 temporary view 36
 temporary views 202
 text 210
 theme 163, 204
 themes 10, 200
 external 18
 thumbnail 124
 TIF 18
 time 98
 TMP 46
 Toolbar 103
 Object Inspector 12
 Object List 157
 Toolbar Designer 103
 TRACE 56
 disable 51
 Trace Debugger
 Data Dictionary 187

 Error List Toolbar 187
 watch variables 187
 transparent 134
 Tree View
 copy 106
 paste 106
 triggers 91

- U -

UNC path 46
 Unicode 184
 UNION 207
 unique key 8, 208
 UNLOAD 14, 55, 203
 data 69
 structure 69
 structure and data 69
 table definition 69
 unselect all 143
 UPDATE 7
 user interface
 notifications 56
 UTF-8 184
 Utilities
 Find in Files 77
 utility 22

- V -

validate email address 80
 validate user input 142
 VARBIT 18
 VARCHAR 18
 variable 19, 209
 global 54
 Variable Date/Time Picker 102
 Variable Edit 98, 110
 balloon tip 121
 cue banner 121
 hint 121
 watermark 121
 Variable Email Label
 BCC 95
 CC 95
 Message Body 95
 Variable Form

Variable Form
 convert 105
Variable Lookup List Box 143, 154
Variable Lookup List View 109, 115, 143, 154
 highlight 115
 multi-select 156
 selected row colors 115
 Zebra Stripes 158
Variable Spinner 154
variables
 ampersand 212
 dotted 212
Variables Image 153
Version 144
view
 temporary 119
views 45

- W -

Web Browser 147
WHERE Builder 32, 69
WHERE Clause 69
WHILE 215
WIDENOTE 19
WIDETEXT 19
wildcard 150
Windows
 default printer 207
 Explorer 207
 print dialog 207
 search 207

- Z -

zoom 41

Back Cover