

# R:BASE Editor 11



## Help Manual



# **R:BASE Editor 11**

## **Help Manual**

---

*by R:BASE Technologies, Inc.*

*Welcome to R:BASE Editor 11 for Windows*

*R:BASE Editor is a stand alone text editor specifically designed for the creation and editing of R:BASE program and application files, command files, and text files.*

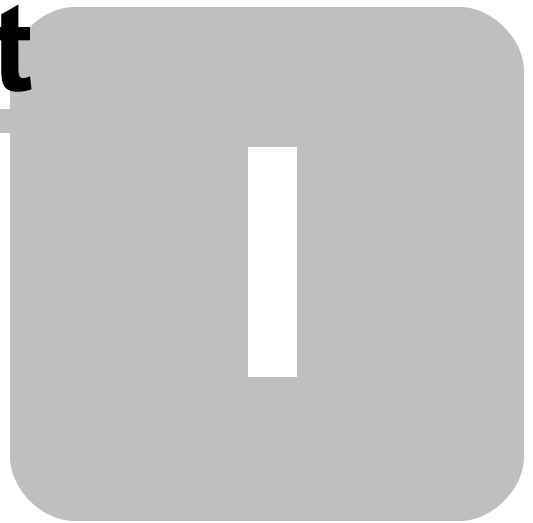
# Table of Contents

<b>Part I Introduction</b>	<b>5</b>
1 Introducing R:BASE Editor 11 .....	6
2 What's New .....	6
3 Copyrights .....	10
4 License .....	11
5 System Requirements .....	14
6 Installation .....	14
7 Complimentary Support .....	15
<b>Part II R:BASE Editor 11</b>	<b>16</b>
1 Menu Bar .....	17
2 Toolbars .....	19
File .....	19
Edit .....	20
Bookmarks .....	20
Settings .....	20
Search .....	21
Insert .....	21
Run .....	21
Calculator .....	21
Character Map .....	22
Clipboard History .....	23
Compare by Content .....	24
Current Directory .....	25
Error Messages .....	26
File Picker .....	26
Structure .....	27
3 Multi-View Tabs .....	29
4 Hot Keys .....	30
5 Code Templates .....	31
6 Command Builder .....	32
7 Data Dictionary .....	34
8 Find in Files .....	35
9 Find Text .....	36
10 Replace Text .....	37
11 Key Assignments .....	38
12 Edit Syntax Scheme .....	38
13 Shell Integration .....	44
14 Spell Checker .....	45

15	Thesaurus .....	46
16	File Encryption .....	47
17	Using Regular Expressions for Finding Text .....	48
<b>Part III Editor Settings</b>		<b>50</b>
1	General .....	51
2	String .....	52
3	Display .....	53
4	Line Highlight .....	55
5	Background .....	55
6	Printer Font .....	57
7	Compare by Content .....	57
8	Structure Toolbar .....	58
<b>Part IV R:Style</b>		<b>60</b>
1	Features .....	62
2	Why use R:Style? .....	63
3	Configuring R:Style .....	64
4	A Bit of Advice .....	68
5	Running R:Style .....	68
6	Order of Processing .....	68
7	Troubleshooting .....	68
<b>Part V Technical Support</b>		<b>71</b>
<b>Part VI Useful Resources</b>		<b>73</b>
<b>Part VII Feedback</b>		<b>75</b>
<b>Index</b>		<b>77</b>



**Part**



# 1 Introduction

## 1.1 Introducing R:BASE Editor 11

R:BASE Editor is a stand alone text editor designed for the creation and editing of R:BASE application files, command files, and text files. This stand-alone text editor is similar to the text editor included with R:BASE, only offers many additional features.

R:BASE development is made easier with [R:Style](#) 11, which provides structural error checking, and spelling assistance to help debug your code prior to running it. R:Style also helps find many errors which cause syntax to fail, and displays where in the file to look.

The Data Dictionary utility, which retrieves database information (table names, column names, etc.), allows for objects names to be inserted into code for accurate syntax writing.

The file Structure Toolbar supports advanced settings to control what commands are displayed, allowing an intuitive breakdown of a command file, and fast locating of code blocks.

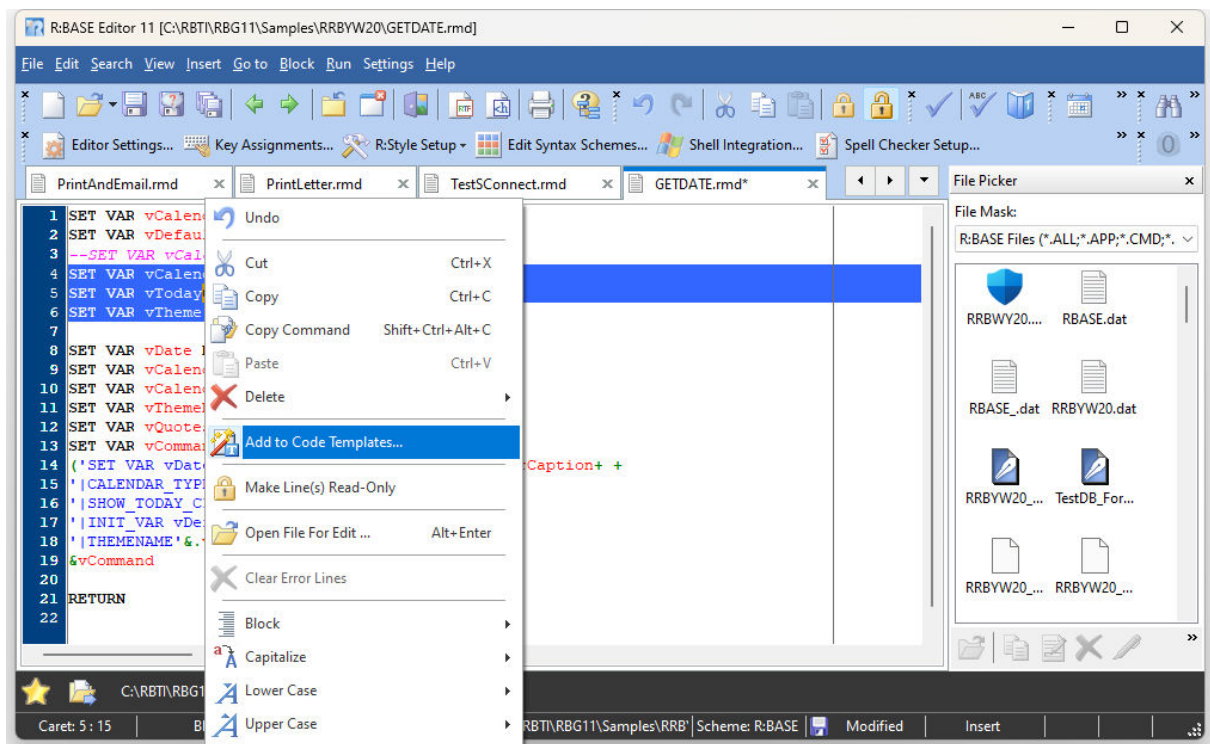
R:BASE Editor includes a 512-bit file encryption, along with other built-in utilities such as the spell checker, thesaurus, and [shell integration](#) to associate your favorite file extensions.

R:BASE Editor can be customized for maximum efficiency with many environment settings, syntax formatting for over thirty programming languages, and over twenty "look and feel" themes.

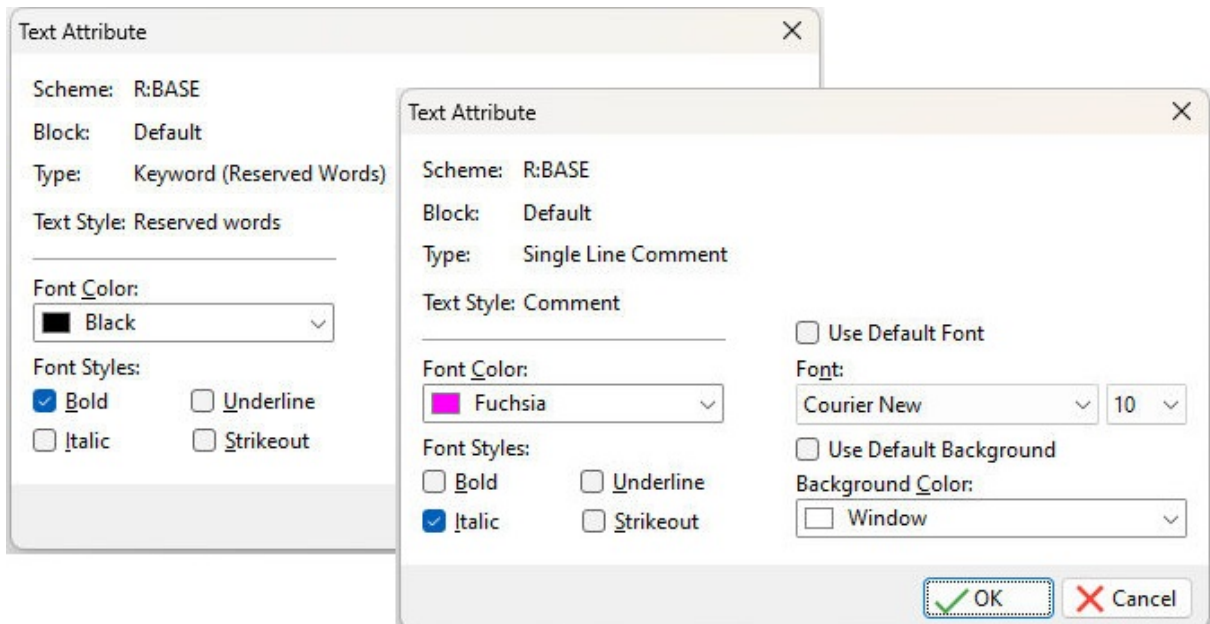
## 1.2 What's New

The R:BASE Editor provides enhanced functionality for working with R:BASE command files. The updated editor offers these special features:

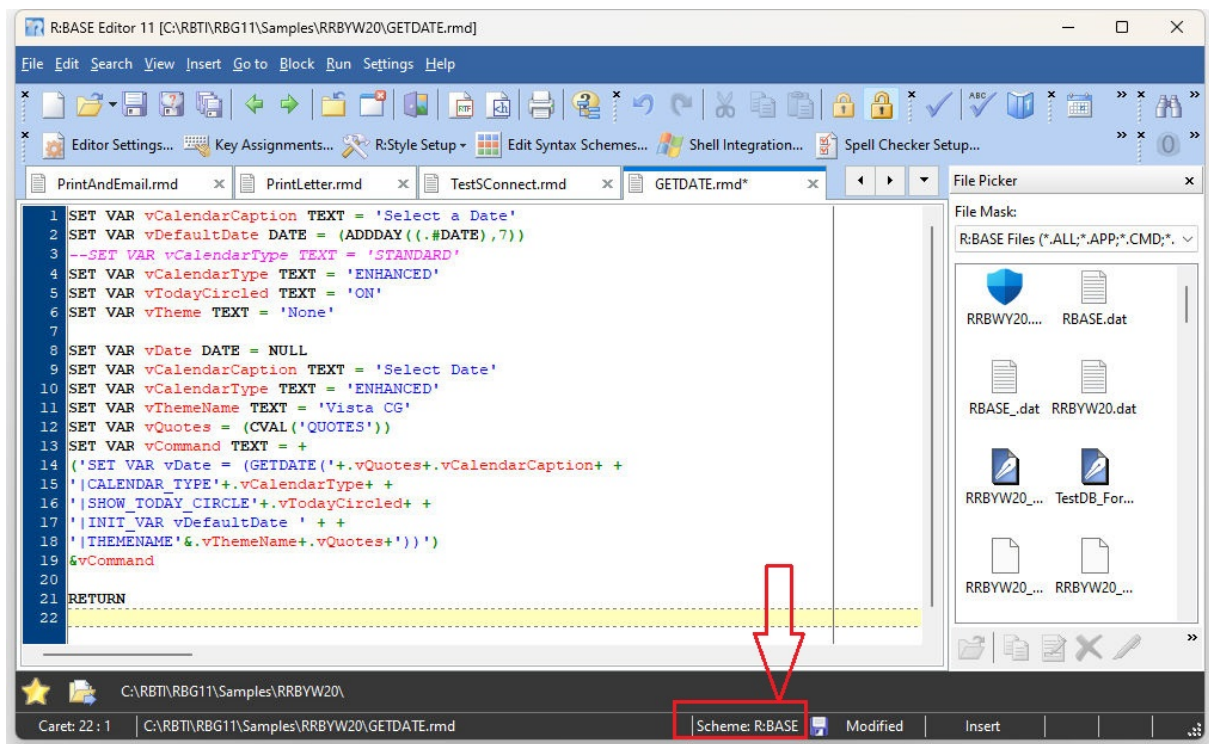
- Added right click "Add to Code Templates" context menu for highlighted text, where a command, or command block, is easily added to the [Code Templates](#) utility. The "Code Templates" dialog will appear with the Name and Description fields blank, and the selected code already in the "Code Template" field.



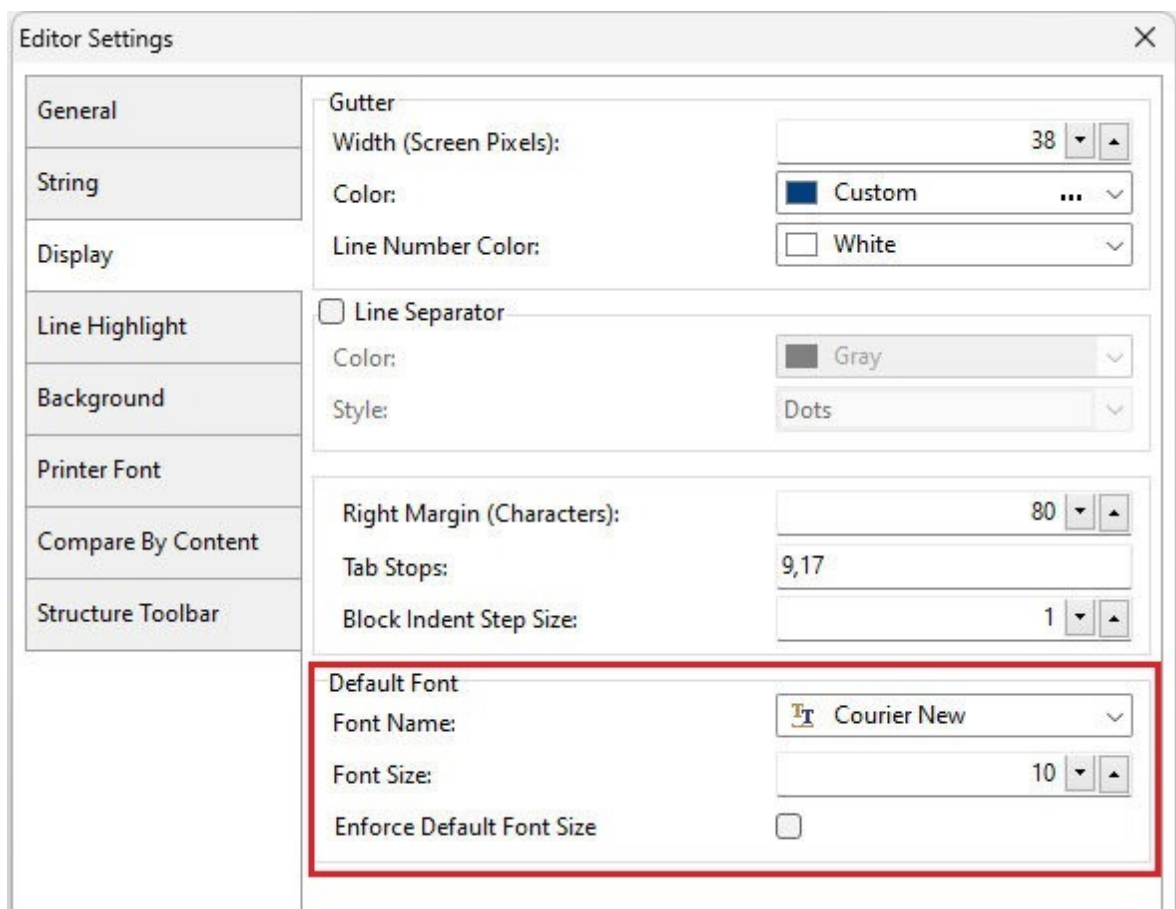
- Ability to review and edit the text attributes for the Syntax Scheme in use (based upon the cursor position), when the [F6] hot key is pressed



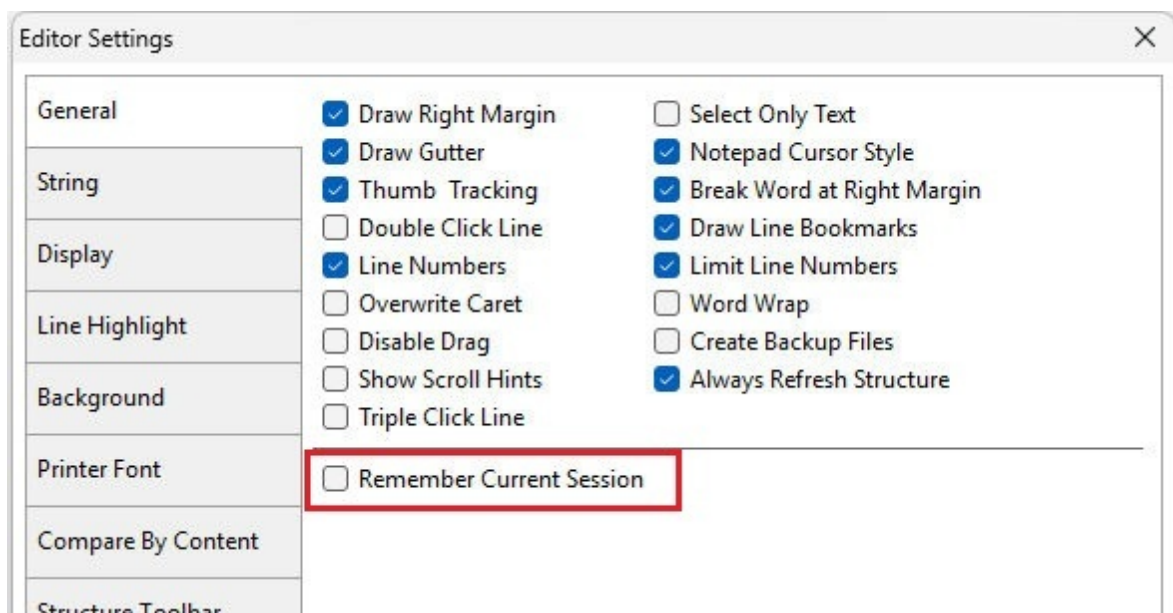
- Added status bar pane showing the Syntax Scheme (R:BASE, Java, MS-DOS, etc.) in use. If the pane is double clicked, the "Scheme Editor" dialog is opened.



- Enhanced file launch capability within the Compare by Content utility by double clicking the file content
- Added display for the current Insert/Overwrite key status in the status bar
- Added "Default Font" menu to the R:BASE Editor Display settings to assign a font type and size to all word types (comments, strings, numbers, identifiers, keywords)



- New ability to save/load an R:BASE Editor Session File (\*.rbf) where a list of command files are opened from a prior session. The Save Session and Load Session options are available from [File](#) menu bar option.
- Added "Remember Current Session" setting to restore a group of commands that were opened in the prior session



## 1.3 Copyrights

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®, RBAAdmin®, R:Scope®, R:Mail®, R:Charts®, R:Spell Checker®, R:Docs®, R:BASE Editor®, R:BASE Plugin Power Pack®, R:Style®, RBZip®, R:Mail Editor®, R:BASE Dependency Viewer®, R:Archive®, R:Chat®, R:PDF Form Filler®, R:FTPClient®, R:SFTPClient®, R:PDFWorks®, R:Magellan®, R:WEB Reports®, R:WEB Gateway®, R:PDFMerge®, R:PDFSearch®, R:Documenter®, RBInstaller®, RBUpdater®, R:AmazonS3®, R:GAP®, R:Mail Viewer®, R:Capture®, R:Synchronizer®, R:Biometric®, R:CAD Viewer®, R:DXF®, R:Twain2PDF®, R:Scheduler®, R:Scribbler®, R:SmartSig®, R:OutLink®, R:HASH®, R:JobTrack®, R:TimeTrack®, R:Manufacturing®, R:QBDataDirect®, R:QBSynchronizer®,

and R:QBDBExtractor®, 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 11-10, Windows Server 2025-2016, Azure Maps, Word, Excel, Access, SQL Server, and Outlook are registered trademarks of Microsoft Corporation. OpenOffice is a registered trademark of the Apache Software Foundation.

Printed: July 2025 in Murrysville, PA

First Edition

## 1.4 License

### R:BASE TECHNOLOGIES, INC. LICENSE AGREEMENT

#### R:BASE Editor 11 Single Seat License

This is a legal agreement between you, the end user ("**Licensee**"), and R:BASE Technologies, Inc. ("**RBTI**"). Please read the terms and conditions of this License Agreement before using this software. By you selecting "I accept the license agreement" and clicking "Next" during product installation means you expressly accept the terms and conditions of this Agreement. If you do not accept the terms and conditions of this Agreement, you must stop installing the Product and click "Cancel". Your money will be refunded based upon proof of purchase, and in compliance with the return period described in the LIMITED WARRANTY below. A violation of the License, brings damage both financially and to the reputation of RBTI, and in the occurrence of either, both termination of the license agreement and civil damages will be vigorously sought. Once you have clicked "I accept the license agreement", you are entitled to use the Product under the following terms and conditions of this Agreement:

#### **LICENSE**

This RBTI License Agreement permits you to use one copy of the R:BASE Editor computer software with associated utilities (the "**Program**") and accompanying user documentation (the "**Documentation**") on any single computer provided the Program is being used on only one computer at a time. If you do not have a 5 Seat Add-On License, this program can only be installed on a single machine. The 5 Seat Add-On License permits additional seats to access the Program. 5 Seat Add-On Licenses must be licensed at the same Single Seat version. A Program is "being used" on a computer when it is loaded into a temporary memory or installed on a hard drive in the computer. Each seat having access to the Program must have an appropriate license. If you anticipate that the number of seats with access to the Program will exceed the number of seats for which you are licensed, you must take steps to ensure that the appropriate licenses are obtained for each seat.

#### **COPYRIGHT AND RESTRICTIONS**

RBTI retains full ownership rights in the Program and Documentation. You may make a single copy of the Program; to be used solely for backup or archival purposes, or you may transfer the Program onto a single hard disk provided you keep the original solely for backup or archival purposes. Such copies shall be owned by RBTI. You may not copy any printed material or Documentation without prior permission. You may not decompile, disassemble cross-compile, reverse engineer, or make or distribute any other form of, or derivative work from, the Program. You may not obscure, alter or remove any RBTI copyright, trademark or proprietary rights notices.

#### **TRANSFER OF LICENSE**

You may not lend, rent or lease the Program or Documentation or any copies to any person. A transfer of license is only authorized through completion of the Consent to Assignment and Assumption of Contract agreement from RBTI. Upon license transfer, you are permanently giving such person possession of all copies of the Program and Documentation, are permanently giving up your right to use the Program and Documentation, and the recipient agrees to the terms of this License Agreement.

#### **STEP UPS AND UPGRADE PURCHASES**

As the Licensee, you are authorized to use the Program only if you are an authorized user of a qualifying product as determined by RBTI. The new license agreement takes the place of the agreement of the qualifying software you stepped up or upgraded from. After you upgrade, you may no longer use the

software from which you upgraded. When you install the upgrade, you must uninstall the copy of the qualifying product.

#### **TERM OF LICENSE**

The licensing provided in this License Agreement is perpetual unless you violate any of its terms or conditions, at which time the license will automatically terminate. Upon termination you must return all copies of the Program and Documentation to RBTI or certify in writing to RBTI that all such copies have been destroyed and uninstalled from each workstation and/or network server. RBTI reserves the right to, at its expense and without prior notice, conduct periodic inspections for licensing compliancy. If licensee is found to be in violation of current agreement, RBTI may commence a civil action seeking fines, damages, attorney's fees and injunctive relief and may also, in appropriate circumstances, seek criminal prosecution.

#### **SOFTWARE SUPPORT**

The availability of software support services is subject to the End of Support (EOS) and End of Life (EOL) product life cycle, and to an active Software Assurance Plan. Where applicable, licensees will be provided with an option to upgrade to the current supported version of a software product.

#### **LIMITED WARRANTY**

RBTI warrants to you, as the initial user, the Program will perform substantially in accordance with the Documentation, provided it is used in unaltered form with functioning equipment and operating systems for which it was designed. RBTI will, at its option, with proof of payment within 30 days of the invoice date and after the Program with Documentation has been deactivated and uninstalled, and software installer, whether provided by download or other means, permanently deleted from all drives and folders, issue a full refund. These are your sole remedies for any breach of warranty. No exceptions will be made.

#### **SERVICES PROVIDED WITH PURCHASE**

##### **1. 30-Day Limited Complimentary Technical Support**

###### **LICENSEE RESPONSIBILITIES**

- To help us expedite the process and provide high quality assistance, the licensee must provide proof of purchase when calling. Proof of purchase is defined as the following: registration number, purchase date, version and build number, and company or individual to which product is registered.
- To have operating system, workstations, and local network installed and functional. RBTI will NOT be responsible for resolving issues not pertaining to the Program.
- Our support staff deals with advanced issues, therefore the person contacting RBTI for assistance should be the system administrator or have other R:BASE/SQL experience and be able to understand and implement the advice given.

###### **R:BASE TECHNOLOGIES, INC. RESPONSIBILITIES**

- To provide quality assistance in a timely manner to aid Licensee in the installation of the product within 30 days of the date of purchase.
- To provide a reasonable solution for any resolvable issue. Not all issues are resolvable, and therefore we will acknowledge the existence of known issues or "bugs" which we are presently aware of, that have no reasonable work-around.

RBTI reserves the right to limit the amount of support time allotted to a maximum of 2 HOURS during the 30-Day Complimentary Technical Support Period. We also reserve the right to limit the quantity of calls from a particular Licensee to 30 MINUTES in a single day. Issues are dealt with on a case-by-case basis, and are handled at the discretion of the support agent assigned to the case. Complimentary Support is limited to **INSTALLATION** and **ELEMENTARY CONVERSION** related issues **ONLY**. Our support hours are from 10am. to 6pm. Eastern Time.

##### **2. Fixes for Known Issues**

RBTI will provide continued product fixes for known issues or "bugs" for 1 YEAR from the software purchase date. After 1 year, users are urged to acquire the necessary Software Assurance Plan to continue with product support and ongoing activations for reinstallations and license transfers.

##### **3. Enhancement Requests**



RBTI will provide continued product enhancements for requested features for 1 YEAR from the software purchase date. After 1 year, users are urged to acquire the necessary Software Assurance Plan to continue with product support and ongoing activations for reinstallations and license transfers.

**NO OTHER WARRANTIES**

Except as explicitly stated above, RBTI makes no express or implied warranties (including any warranties of merchantability or fitness) with respect to the character, function, or capabilities of the program, the documentation or their appropriateness for any user's purposes. RBTI cannot customize product(s) to meet specific needs in all cases. Examples of customization include, but are not limited to: special character sets, foreign language adaptations, specific device drivers or other localization issues. Under no circumstances will RBTI be held responsible for product functionality once alterations have been made to accommodate individual needs.

**DISCLAIMER OF WARRANTY**

The Program and the accompanying files are sold "as is" and without warranties as to performance or merchantability or any other warranties whether expressed or implied. Because of the various hardware and software environments into which the Program may be put, No warranty of fitness for a particular purpose is offered. Good data processing procedure dictates that any program be thoroughly tested with non-critical data before relying on it. The user must assume the entire risk of using the Program. Any liability of the seller will be limited exclusively to product replacement or refund of purchase price.

**HIGH RISK ACTIVITIES**

The Program is not fault-tolerant and is not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, air traffic control, aircraft navigation or communication systems, direct life support machines, or weapons systems, in which the failure of the Program could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). RBTI specifically disclaims any expression or implied warranty of fitness for High Risk Activities.

**LIMITATIONS ON LIABILITIES**

RBTI will not be responsible for any costs or damages associated with loss of the use of the Program or any other resources, loss of business or profits, any loss of data, any third-party claims or costs of substitute programs. In no event will RBTI be liable for any incidental, indirect, special, consequential or punitive damages suffered by the user or any other person or entity, whether from the use of the program or documentation, any failure thereof, or otherwise, even if RBTI or its dealers or agents are aware of the possibility of such damages. In no event will RBTI aggregate liability to you or anyone else exceed two times the license fee you paid for the program and documentation in this package. Because some states do not allow the limitation or exclusion of implied warranties and liabilities for consequential or incidental damages, the above limitations may not apply to you.

**GENERAL**

This License Agreement constitutes the full and complete agreement between parties. RBTI retains all rights not specifically granted herein. RBTI shall not be deemed to have waived any of its rights hereunder or under all copyright laws, trade secrecy laws or otherwise. This Agreement is intended as a legally binding agreement which will be enforced to the full extent permitted under applicable law, in whole or in part. If any one provision of this Agreement is declared invalid or unenforceable, all remaining provisions shall never less remain in effect. The laws of the state of Pennsylvania shall govern this Agreement. RBTI reserves the right to amend, alter, or revoke this agreement at any time. All revisions to this license agreement are available for inspection upon request, supersede conditions in all past agreements, and render prior license agreements void and unenforceable. Both parties named in this license will only be bound to the terms of the most current revision of this agreement.

**U.S. GOVERNMENT RESTRICTED RIGHTS**

Use, duplication or disclosure by the Government, its agents or employees is subject to all restrictions imposed by law, regulation or government directive, including but not limited to those restrictions set forth in DFARS 252.227-7013 and 48CFR 52.227-19, as applicable.

R:BASE Technologies, Inc.  
<https://www.rbase.com>  
[rbaseinfo@rbase.com](mailto:rbaseinfo@rbase.com)

Copyright 1982-2025 R:BASE Technologies, Inc.  
All Rights Reserved

Revised Monday, July 07, 2025

## 1.5 System Requirements

The following system specifications are recommended for the optimal use of R:BASE and R:BASE-related software.

### Workstation Hardware

- 2-Core 2GHz+ CPU
- 2 GB of available RAM (4 GB recommended)
- 2 GB of available hard disk space
- 1024x768 or higher resolution video adapter and display
- Standard mouse or compatible pointing device
- Standard keyboard

### Server Hardware

- 2-Core 2GHz+ CPU
- 6 GB of available RAM (8 GB recommended)

### Operating System

- Microsoft Windows 11 (Professional)
- Microsoft Windows 10 (Professional)
- Microsoft Windows Server 2025
- Microsoft Windows Server 2022
- Microsoft Windows Server 2019
- Microsoft Windows Server 2016

### Network

- Ethernet infrastructure (Gigabyte recommended)
- Internet connection recommended, but not required, for license activation, software updates, and support
- Anti-virus programs should exclude the R:BASE program, and any add-on product, executable and database files

## 1.6 Installation

The installation of R:BASE Editor is fully automated, and does not require user intervention for the initial configuration.

Run the installer ".exe", provided by download, while physically sitting at the workstation to begin the installation process, and read the installer screens for licensing and other information as the program installs.

### Installation Directory

C:\RBTI\REdit

### Files Installed

REdit11.exe  
REdit.pdf  
REdit.chm  
ReadMe.txt  
License.rtf  
COMMAND.INI  
CodeTemplates.rct

### Installation Subdirectories

Dictionaries  
SyntaxSchemes

## 1.7 Complimentary Support

### 30 DAY LIMITED COMPLIMENTARY TECHNICAL SUPPORT

#### A. LICENSEE RESPONSIBILITIES.

1. To help us expedite the process and provide high quality assistance, the licensee must provide proof of purchase. Proof of purchase is defined as the following: registration number, purchase date, version and build number, and company or individual to which product is registered.
2. To have operating system, workstations, and local network installed and functional. R:BASE Technologies will NOT be responsible for resolving issues not pertaining to the software product.
3. Our support staff deals with advanced issues, therefore the person contacting R:BASE Technologies for assistance should be the system administrator or have other R:BASE/SQL experience and be able to understand and implement the advice given.
4. To have the database, application, and command files being reviewed, safely backed-up before attempting assistance. R:BASE Technologies will NOT be held responsible for lost data or corruption as a result of advice given.

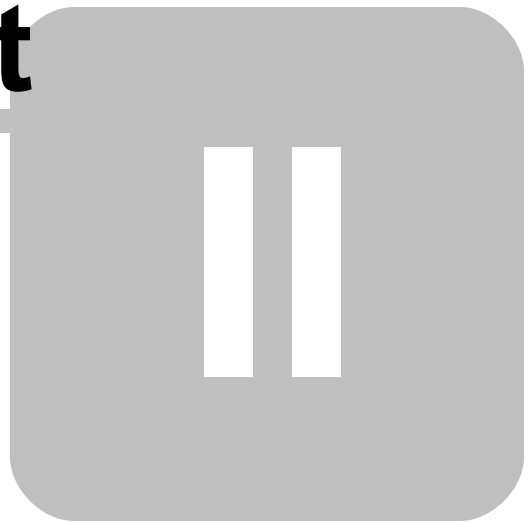
#### B. R:BASE TECHNOLOGIES, INC. RESPONSIBILITIES.

1. To provide quality assistance in a timely manner to aid in the installation of the product and elementary conversion of database, application, and command files within 30 days of the date of purchase.
2. To provide a reasonable solution for any solvable issue. Not all issues may be solved, and therefore we will acknowledge the existence of known issues, or bugs, which we are presently aware of, that have no reasonable work-around.

R:BASE Technologies reserves the right to limit the amount of support time allotted to a maximum of 2 HOURS during the 30-Day Complimentary Technical Support period. We also reserve the right to limit the quantity of calls from a particular licensee to 30 MINUTES in a single day. Issues are dealt with on a case-by-case basis, and are handled at the discretion of the support agent assigned to the case. Complimentary Support is limited to INSTALLATION and ELEMENTARY CONVERSION related issues ONLY. Our support hours are Monday through Friday, from 10:00 AM to 6:00 PM (EST).

For application, design, or advanced conversion assistance, R:BASE Technologies offers Technical Support Plans of various types to meet your needs. Please visit the Support page at <https://www.rbase.com/support> for details and pricing.

**Part**

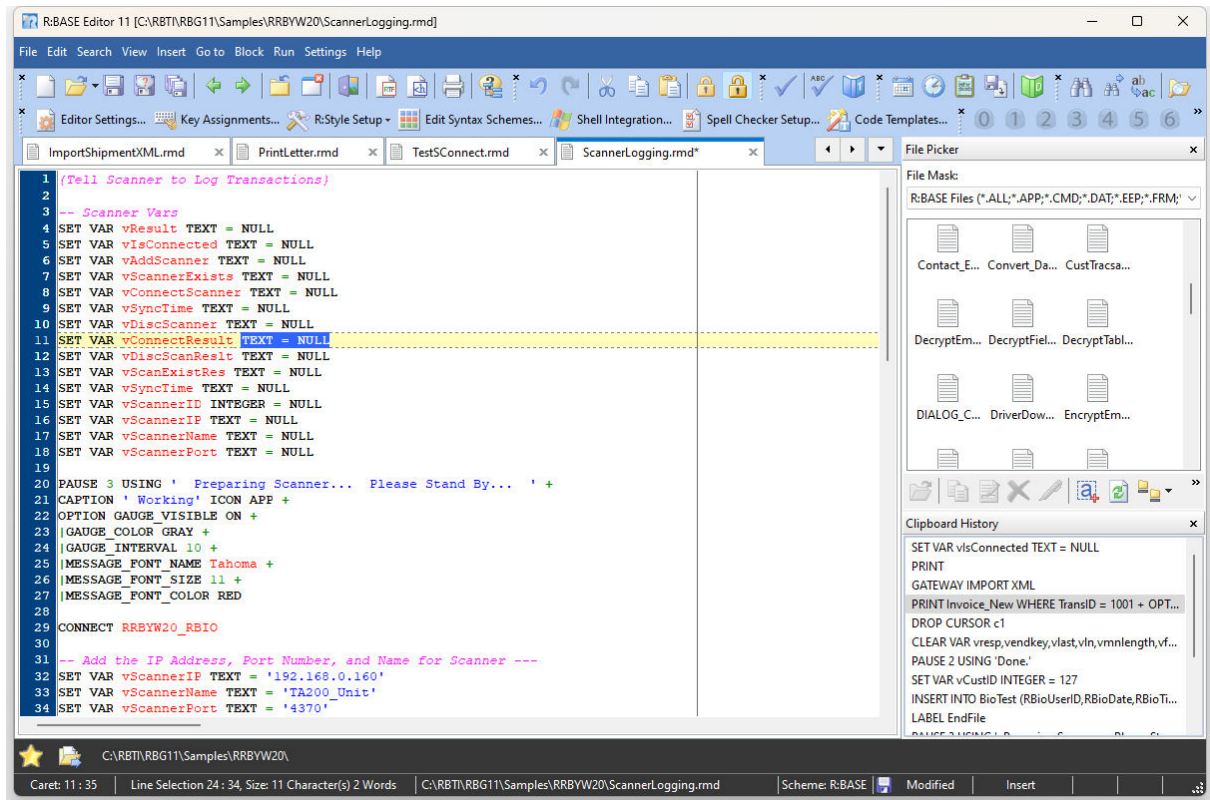


## 2 R:BASE Editor 11

The editor window contains a separate tabbed page for each command file you open. When you start R:BASE Editor, an empty files will open with the word "Untitled" on the tab. You can already begin entering command syntax and save it as a command file. When more than one file is open, you can select the file to work on by clicking on the appropriate tab.

The appearance of the editor window is determined by the syntax scheme highlighting you have selected. You may switch syntax schemes at any time. Right-click anywhere in the editor window and select "Switch Syntax Scheme".

The R:BASE Editor also includes [R:Style](#); which provides structural error checking, and spelling assistance to help debug your code prior to running it.



### 2.1 Menu Bar

#### File

- **New** - allows you to create a new command file
- **Open** - opens an existing command file
- **Insert from File(s)...** - inserts an existing command file in the current opened file
- **Load Session...** - loads a R:BASE Editor session file, to display a group of commands files
- **Save Session...** - saves the list of opened command files as a R:BASE Editor session file (\*.rsbf)
- **File Peek** - displays a read-only view for the first 2MB of large ASCII files, to review the content
- **Information** - displays line, word, and character count statistics for the file. Statistics for selected text is displayed in the status bar.
- **Encoding** - specifies the encoding for the file
- **Save** - saves the current command file
- **Save As** - saves the current command file as a new command file

- *Change Working Directory* - changes the R:BASE Editor working directory
- *Export* - exports the current command file to RTF or HTML format
- *Print Setup* - opens the Print Setup dialog for changing page specific settings
- *Print* - prints the current command file to the default printer
- *Recently opened files* - allows quick access to the last ten opened command files
- *Close Tab* - closes the current command file
- *Close All* - closes all command files
- *Exit* - closes R:BASE Editor

### Edit

- *Undo* - undoes the last change made
- *Redo* - reverses the effect of the last undo
- *Cut* - Cuts the currently selected object(s)
- *Copy* - Copies the currently selected object(s)
- *Copy Command* - Copies the currently selected command and removes any continuation characters (+) so the command can be Pasted at the R>.
- *Paste* - inserts or Pastes the currently Cut or Copied object(s)
- *Make Line(s) Read-Only* - text selection becomes read-only or removes read-only status, for the current session
- *Lower Case* - sets the currently selected text to lower case letters
- *Capitalize* - sets the currently selected text to capitalize the first letter of each word
- *Upper Case* - sets the currently selected text to upper case letters
- *Insert Line* - inserts a line into the current command file
- *Delete Line* - deletes a line from the current command file

### Search

- *Find* - opens Find Text dialog box to locate text in a command file
- *Find Next* - locates next text where matches search
- *Replace* - opens Replace Text dialog box to locate and replace text in a command file
- *Find in Files...* - opens Find File dialog box to locate files by directory, extension, and keyword

### View

- *Toolbars* - displays the list of available tool bars to open and close

### Insert

- *Insert Date* - inserts current date
- *Insert Time* - inserts current time
- *Insert GUID* - generates globally unique identifier (32-character)
- *Insert File Name* - inserts the current path and file name, for documentation purposes
- *List of Commands at Cursor* - displays a command and function list at the cursor location
- *Build Command at Cursor* - displays the Command Builder dialog at the cursor location
- *Data Dictionary* - displays the Data Dictionary for finding R:BASE database information

### Go to

- *Go to Line* - takes the cursor to a specified line number
- *Place Bookmarks* - will create a bookmark at the current cursor location in the file
- *Remove Bookmarks* - will remove a bookmark in the file
- *Go to Bookmarks* - takes the cursor focus to a specified bookmark

### Block

- *Delete* - deletes characters, lines, or words at the current cursor location
- *Jump* - jumps the cursor to beginning or end of a command block, file, line, or word
- *Indent Block* - Indents all lines in the currently selected text
- *Outdent Block* - Outdents all lines in the currently selected text
- *Comment* - converts the current selection to comments using the "double dash" comment syntax
- *Uncomment* - converts the selected text into uncommented code
- *Write Block to File* - copies the currently selected block and places it into a new command file
- *Read Block from File* - copies an existing command file into the current command file
- *Duplicate Block* - duplicates selected block

## Run

- [R:Style](#) - runs the R:Style utility to check and format your code
- [Spell Check](#) - runs the Spell Check utility to check spelling errors
- [Thesaurus](#) - runs the Thesaurus utility to provide possible replacement words

## Settings

- [Editor Settings](#) - presents you with the configuration panel for the [R:BASE Editor](#) settings
- [Key Assignments](#) - allows the creation of custom hot-key assignments and the editing of existing key maps
- [R:Style Setup](#) - allows access to the [R:Style configuration](#) screen (Global or Local) that gives you a great degree of control over how R:Style operates
- [Edit Syntax Schemes](#) - allows you to [Edit Syntax Schemes](#) for existing and new syntax schemes
- [Shell Integration](#) - allows you the operating system [Shell Integration](#) to assign file extensions to the R:BASE Editor program
- [Spell Checker Setup](#) - allows access to the [Spell Checker](#) configuration screen
- [Code Templates](#) - displays the Code Template Editor to load frequently used code fragments

## Help


- [Table of Contents](#) - displays the R:BASE Editor in-line help table of contents
- [Help Index](#) - displays the R:BASE Editor in-line help documentation index
- [Help Search](#) - displays the R:BASE Editor in-line help documentation search
- [R:BASE Home Page](#) - opens the R:BASE Technologies, Inc. home page in your web browser
- [About R:BASE Editor](#) - displays current information about your product such as version

## 2.2 Toolbars

### 2.2.1 File










Icon	Description
	New command file
	Open existing file
	Save file
	Save file as
	Save All
	Moves to the next file
	Moves to the previous file
	Closes file
	Close All
	Closes R:BASE Editor
	Saves as RTF File
	Saves as HTML File
	Print

	Launches the R:BASE Editor Help
---	---------------------------------

## 2.2.2 Edit

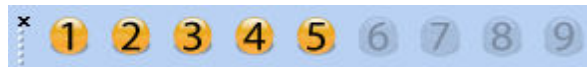


Icon	Description
	Undo changes
	Redo changes
	Cut
	Copy
	Paste
	Makes lines read-only, for the current file instance
	Removes read-only status

## 2.2.3 Bookmarks

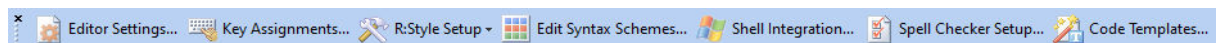
R:BASE Editor bookmarks allows for fast command file navigation by setting up bookmarks at certain file locations and jumping to a bookmark directly. Nine different bookmarks can be added to one file. Bookmarks can be toggled from the R:BASE Editor main menu bar under "Goto" > "Toggle Bookmarks" and also using the [Ctrl+Shift+n] key combination (n = 1-9).

Bookmarks allow users to easily jump from one location to another by clicking on the number within the toolbar, and using the [Ctrl+n] key combination (n = 1-9).



## 2.2.4 Settings

The Settings toolbar opens various environment settings and utilities for R:BASE Editor



Description
<a href="#">Editor Settings</a>
<a href="#">Key Assignments</a>
<a href="#">R:Style Setup</a>
<a href="#">Edit Syntax Schemes</a>
<a href="#">Shell Integration</a>
<a href="#">Spell Checker Setup</a>
<a href="#">Code Templates</a>



## 2.2.5 Search



Icon	Description
	<a href="#">Finds text</a>
	Finds next text
	<a href="#">Replace</a>
	<a href="#">Find in Files</a>

## 2.2.6 Insert



Icon	Description
	Insert date
	Insert time
	Inserts a GUID unique identifier
	Insert file name
	Opens the <a href="#">Data Dictionary</a>

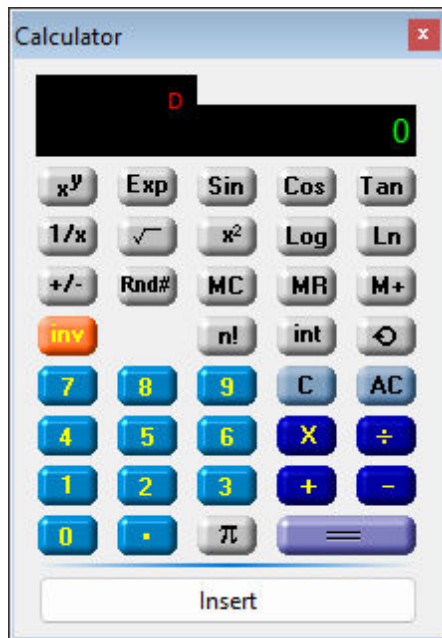
## 2.2.7 Run



Icon	Description
	Runs <a href="#">R:Style</a>
	Runs the <a href="#">Spell Checker</a>
	Runs the <a href="#">Thesaurus</a>

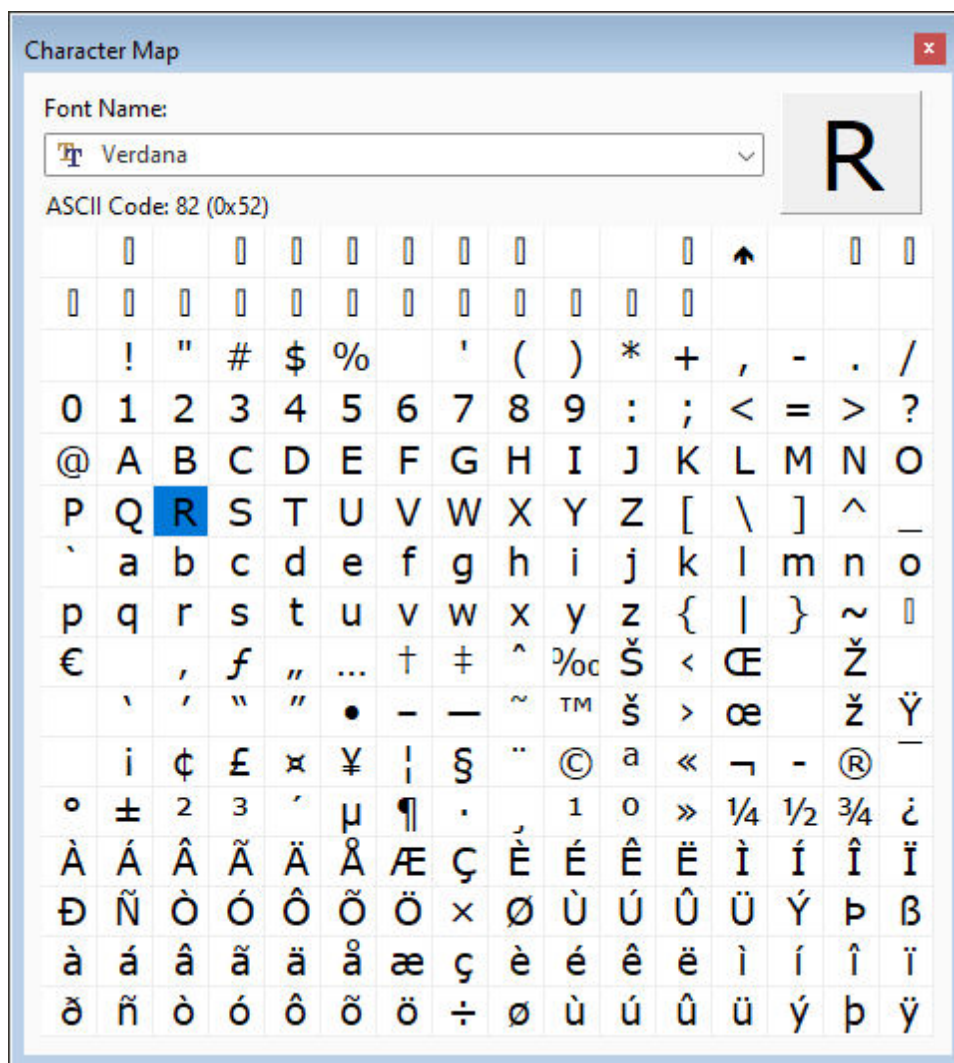
## 2.2.8 Calculator

The Calculator Toolbar allows the display of a calculator interface. The "Insert" button on the bottom of the tool bar will insert the calculator value into your command file.



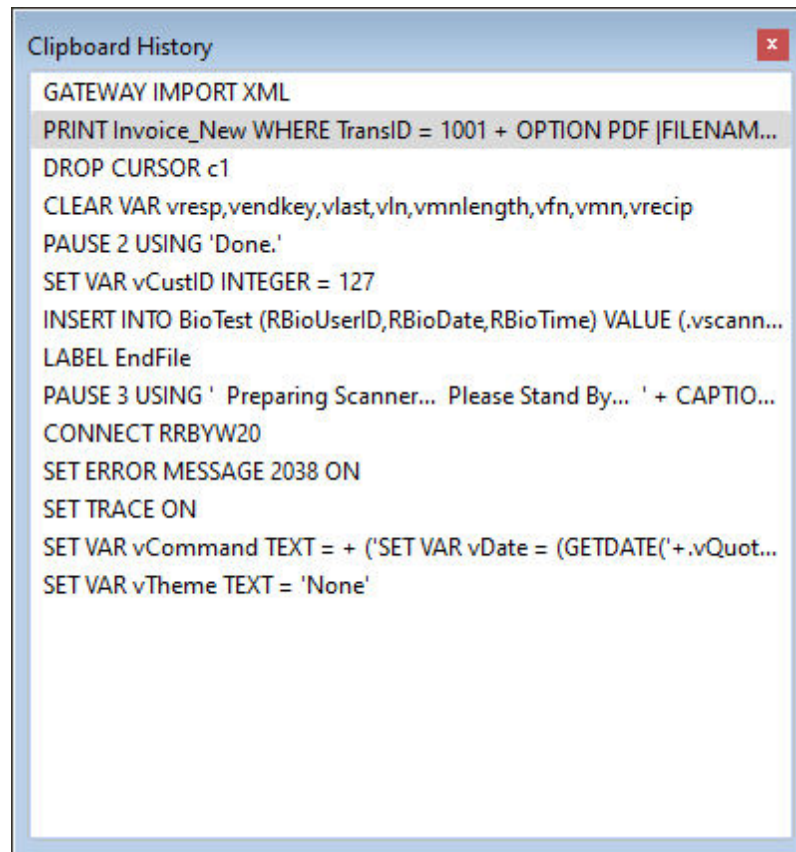
### 2.2.9 Character Map

The Character Map Toolbar will display the ASCII (American Standard Code for Information Interchange) character chart. Selecting a character from the chart will display that character within the preview panel on the top right of the tool bar. A double click on that character or pressing [Enter] will insert the character into your command file. The toolbar height and width may be altered by dragging the border.



### 2.2.10 Clipboard History

The Clipboard History toolbar tracks clipboard operations in R:BASE Editor to be easily reused. The history is per session, and is cleared when R:BASE is closed.

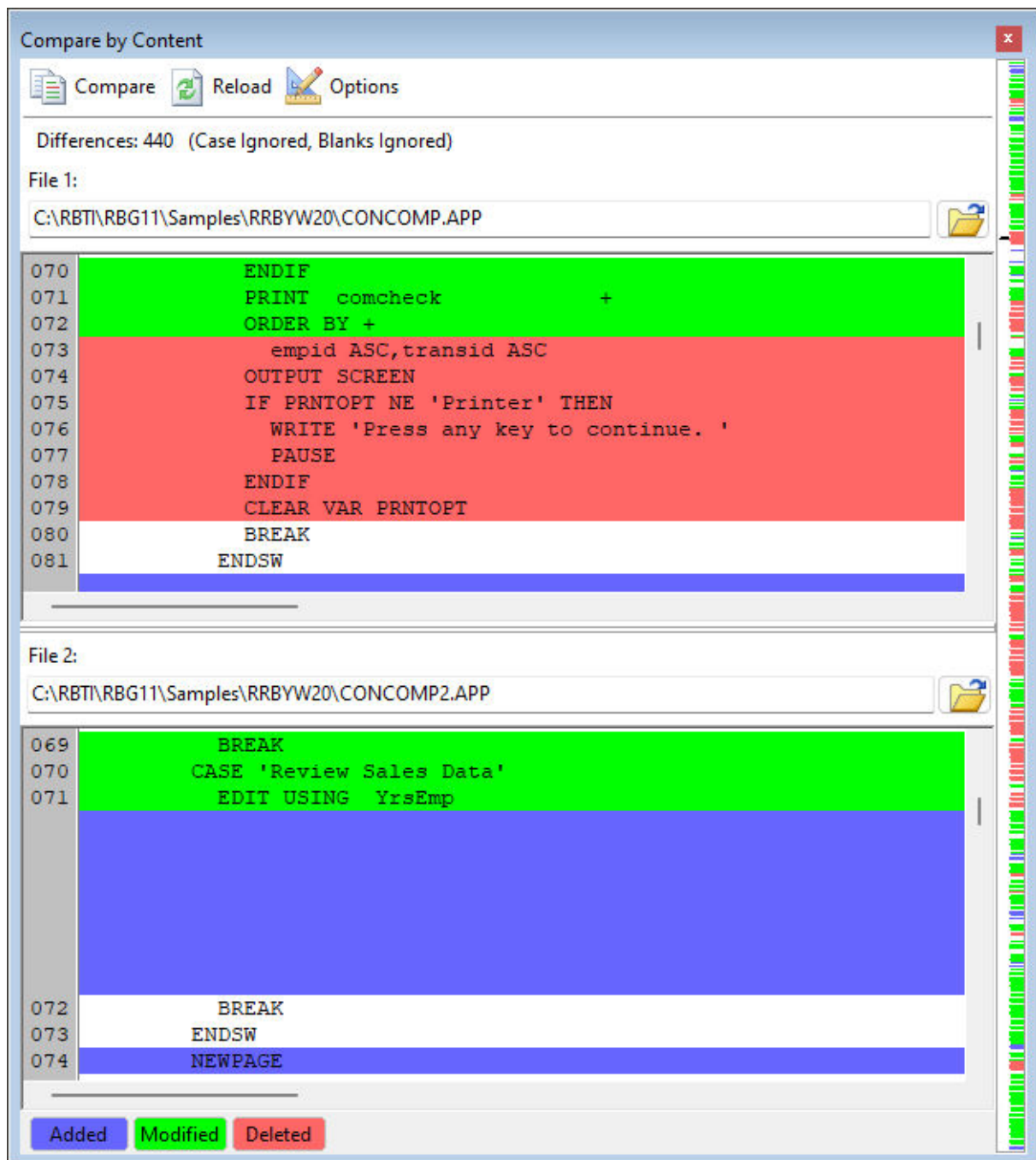


### 2.2.11 Compare by Content

The Compare by Content Toolbar is used to compare two files and to review the command line differences within each. To load files, the open folder button may be clicked to open the current folder location for a file to be selected. File names may also be typed directly into the field, followed by the [Enter] key. Command files displayed as tabs in the editor may also be dragged and dropped (from tab to file name field) to be selected for comparison.

After two files are loaded, select the "Compare" button. File 2 will be compared to File 1 and the differences will be displayed in the form of highlighted lines. The line highlighting will display lines of code which were added to, modified, or deleted. The colors of the highlighting can be altered within the Editor Settings panel under the [Compare by Content](#) tab. The file dates for the two files being compared is also displayed.

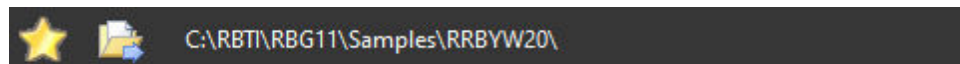
Along the right side of the toolbar window, a vertical bar displays key points in the command file comparison. Colored line markers appear down the vertical key displaying the differences within the files, and a black mark for your current display location in the window.



## 2.2.12 Current Directory

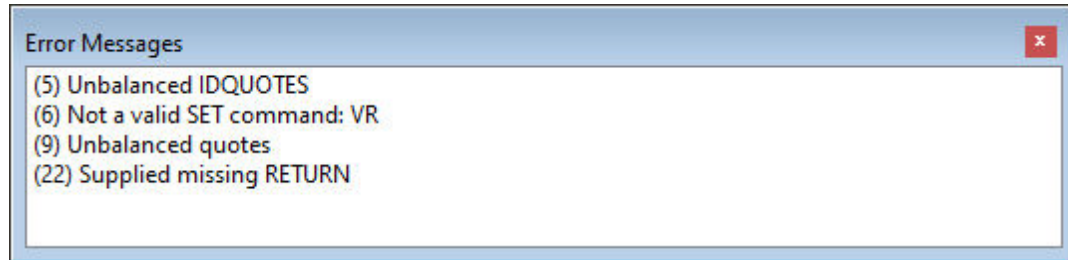
The Current Directory toolbar allows you to set the working directory for the R:BASE Editor instance. It will remember this location after R:BASE Editor is closed and launched later.

The toolbar also includes a "Favorites" button to add and store a list of favorite folder locations that you can easily switch to.



### 2.2.13 Error Messages

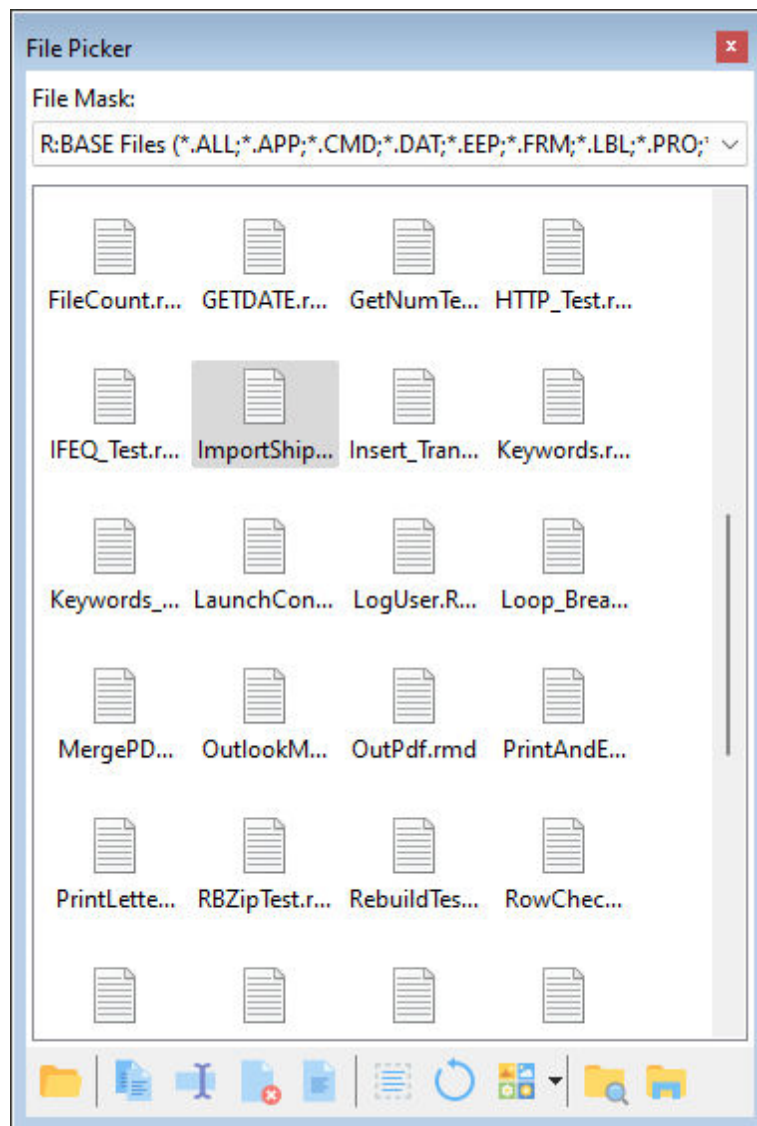
The Error Message Toolbar displays any errors recognized by [R:Style](#) after running the utility.



### 2.2.14 File Picker

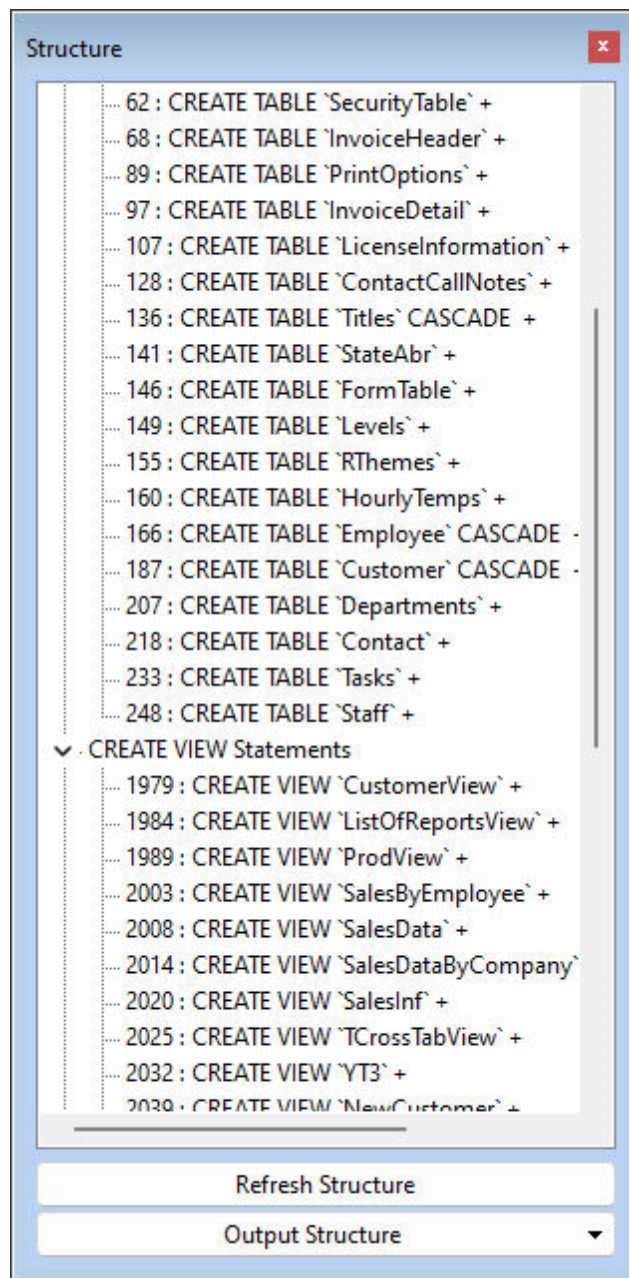
The Files Picker Toolbar contains all the programs files in the directory shown in the "Current Directory Toolbar" for the File Mask specified. You can double click a file in this box to open it in the editor. You can control which files appear in this list using the [Edit Syntax Schemes](#) configuration dialog.

The File Picker window includes a User-Defined Mask. When this mask is populated and applied, only files matching the mask will be displayed in the picker window and with a File | Open. To use, select the User-Defined file mask menu option from the File Picker tool bar, type a valid file mask from the list, then click the red check mark to the right of the window to apply the mask.

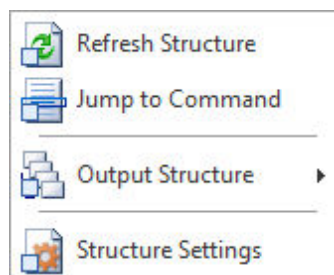


## 2.2.15 Structure

The Structure toolbar displays specific statements and command blocks for the current displayed command file. The commands are presented in a tree format to easily reference the line number where the command instance is located. Double clicking an item in the panel will move the cursor to that position in the command file. Any changes to the structure can be refreshed by selecting the Refresh Structure button. An option is also available to copy the Structure to the Printer, Text File, PDF, or Windows Clipboard.

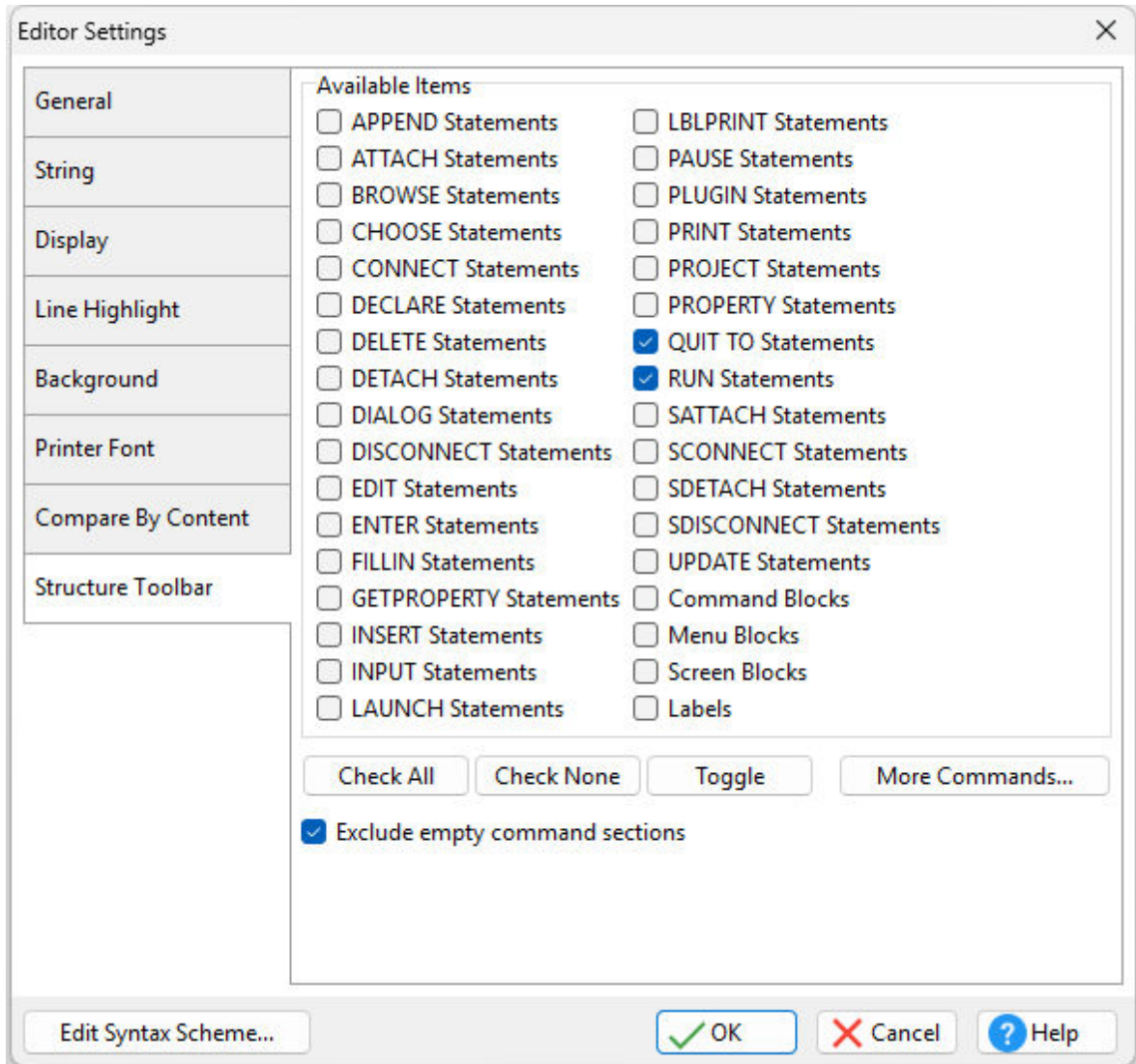


The Structure Toolbar right-click context menu includes refresh, jump to command, various output options, and a settings shortcut.





The statements and commands to display within the Structure toolbar can be modified within the R:BASE Editor Settings.



## 2.3 Multi-View Tabs

When files are opened in the R:BASE Editor, each file is assigned a separate tab. The file name and extension will be listed on the tab, along with a "x" to close that tab.

When more than one file/tab is open, you can view a file by clicking on the appropriate tab. Switching between tabs can also be performed using key combinations to move to the next tab [Ctrl+Alt+N] and previous tab [Ctrl+Alt+P].

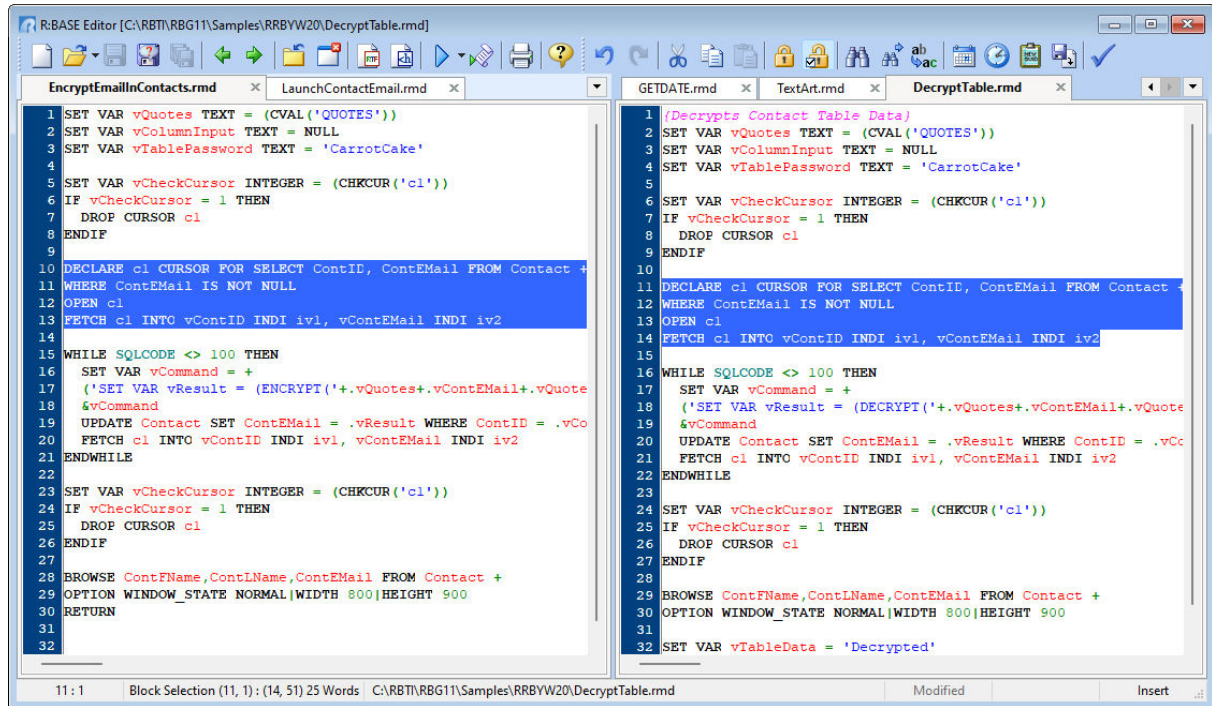
The tabs may be reordered by dragging a tab to the left and right in the tab list. A new file/tab can be created by double clicking in the tab background (empty area to the next to the most far right tab).

### Multi-View Support

With several files open, command files may be displayed side-by-side. To display two files simultaneously, right click on a file tab and select the "Move To Other View" option. The file will be moved

to a new pane. Any newly opened files will be added to the active pane, whether it is the left or right pane. Command file tabs can be moved between the two panes, and each pane can be resized. Tabs may also be dragged to another location/position within a pane.

Synchronized scrolling for moving up and down two files mutually is possible with dual command files displayed simultaneously. Press the [Ctrl] key while scrolling the mouse wheel to activate synchronized scrolling.



## 2.4 Hot Keys

The following hot keys are supported while in the R:BASE Editor:

### Functionality:

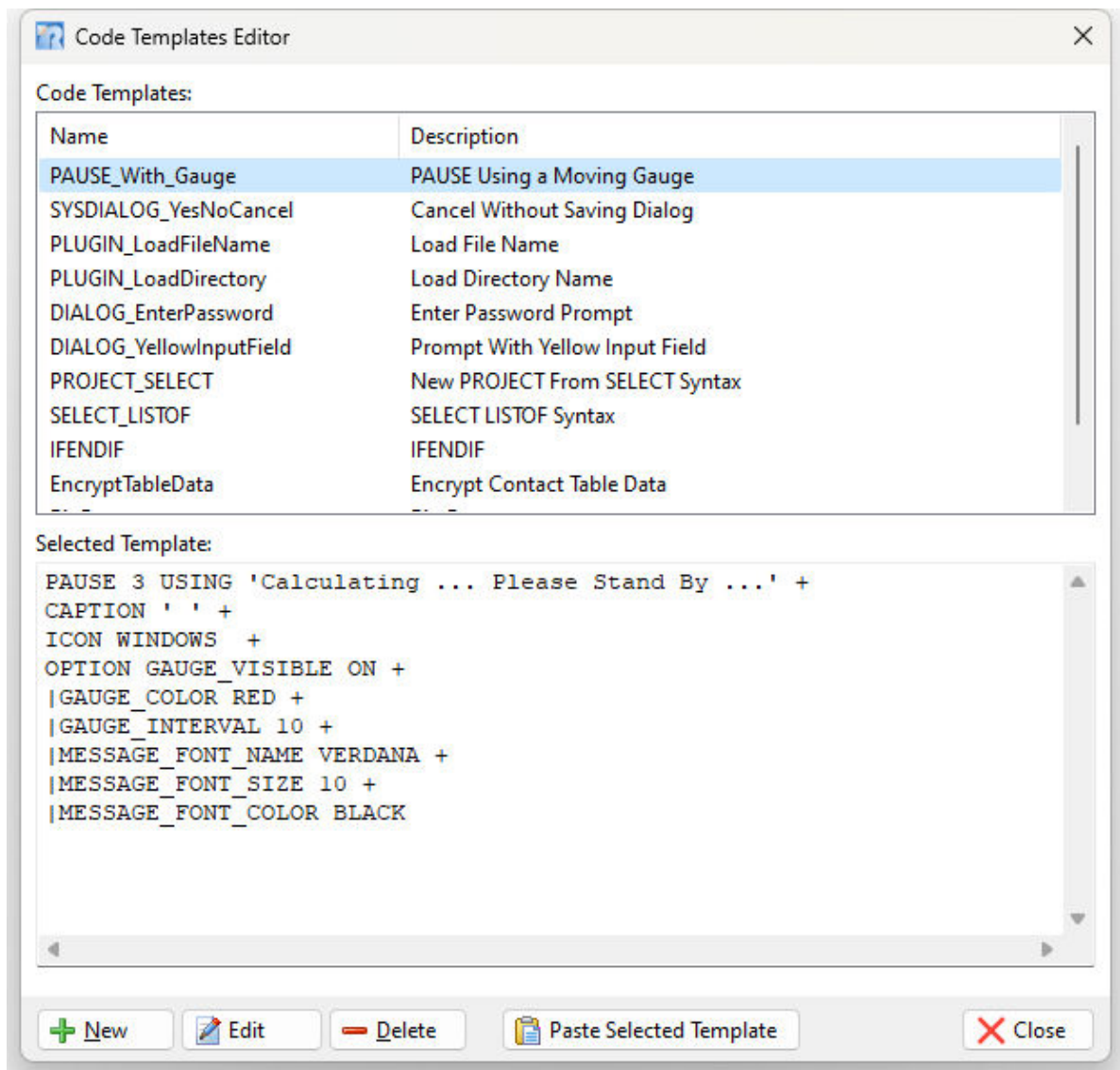
[F4]	Go to Line Number
[F5]	Displays a pop up of command and function templates
[F6]	Displays text attributes for the <a href="#">Syntax Scheme</a> in use
[F8]	Find Next
[F9]	Delete Line
[F10]	Insert Line
[Ctrl+F5]	Displays the <a href="#">Command Builder</a> with command and function templates
[Ctrl+F8]	Run Selected Code
[Ctrl+F9]	Run File
[Ctrl+n]	Goto Bookmark n (n = 1-9)
[Ctrl+A]	Select All
[Ctrl+C]	Copy
[Ctrl+F] or [Shift+F8]	Find
[Ctrl+I]	Indent Block
[Ctrl+N]	Create New File

[Ctrl+O]	Open File
[Ctrl+P]	Print File
[Ctrl+S]	Save File
[Ctrl+U]	Outdent Block
[Ctrl+V]	Paste
[Ctrl+X]	Cut
[Ctrl+Y]	Run R:Style
[Ctrl+Z]	Undo
[Ctrl+Alt+G]	Launches R:Style Setup configuration (global)
[Ctrl+Alt+J]	Inserts code, based upon the Code Template name
[Ctrl+Alt+L]	Launches R:Style Setup configuration (local)
[Ctrl+Alt+N]	Next Tab
[Ctrl+Alt+P] [Ctrl+Alt+.]	Previous Tab
[Ctrl+Alt+Shift+C]	Copy Command (without continuation characters)
[Ctrl+Shift+n]	Toggle Bookmark (n = 1-9)
[Ctrl+Shift+J]	Launches the <a href="#">Code Templates Editor</a>
[Ctrl+Shift+F9]	Trace File
[Ctrl+Up Arrow]	Scroll Up
[Ctrl+Down Arrow]	Scroll Down
[Shift+F1]	Launches the R:BASE Editor Help
[Shift+F7]	Find and Replace
[Shift+F3]	Command/Function Sensitive Help

\* Use the [Ctrl+Alt+Shift+C] key combination to copy the highlighted command to windows clipboard. If the copied command wraps over multiple lines with continuation characters, the pluses "+" will be removed at the end of each line. This allows users to easily paste [Ctrl+V] the command at the R> Prompt to execute and test the command. This is different than the regular copy command which will not remove the "+" at the end of each line when pasted at the R> Prompt.

## 2.5 Code Templates

Code Templates allow for programmers to easily load frequently used code fragments. A Code Template stores a description, block of code, and a short name, which is specifically typed during coding to be expanded to the stored code template with the [Ctrl+Alt+J] key combination. The block of code for templates also preserve indents for multi-line commands, just as they were stored.



In addition to loading the template's block of code, the cursor location can be determined by inserting a carat character (^) into the template source, allowing immediate typing for programming efficiency. For example, a code template storing a DIALOG command can have a carat placed at the "message" parameter location, so the custom message can be immediately defined after the block of code is loaded.

The Code Templates are stored in the CodeTemplates.rct file, which should always be located in the local <UserProfile>\RBTI\ folder.

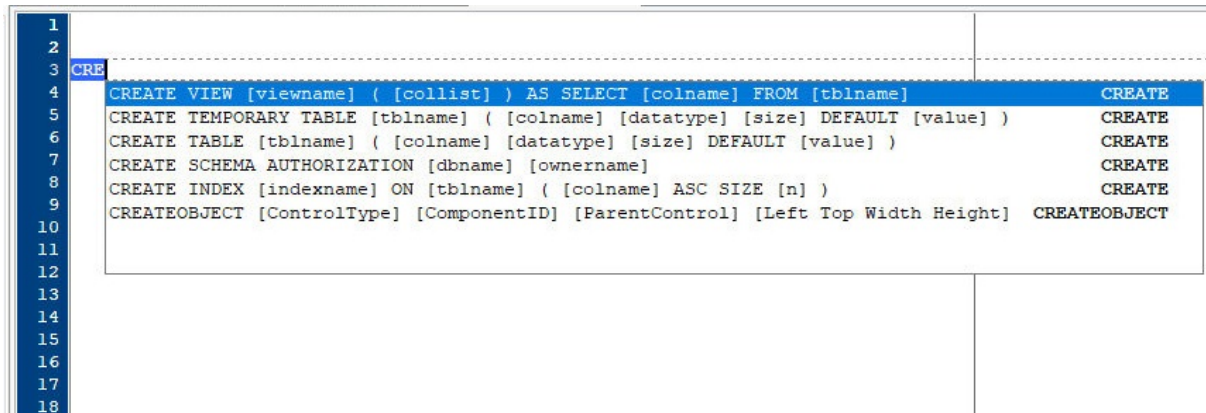
## 2.6 Command Builder

The R:BASE Editor allows for command and function templates to be inserted at the current cursor location while coding in command files, custom EEPs, custom form actions, etc. coding areas. The list of commands and functions are loaded from the COMMAND.INI file within the R:BASE Editor program folder.

### List of Commands at Cursor

Pressing the [F5] key will display a pop up of commands and functions, sorted alphabetically, at the cursor location. Entering a few letters for the desired command or function before pressing [F5] will limit

the results of the popup. Selecting an item in the popup will place that template in the coding area. The inserted template can then be modified for use within the program.



### Comment Templates

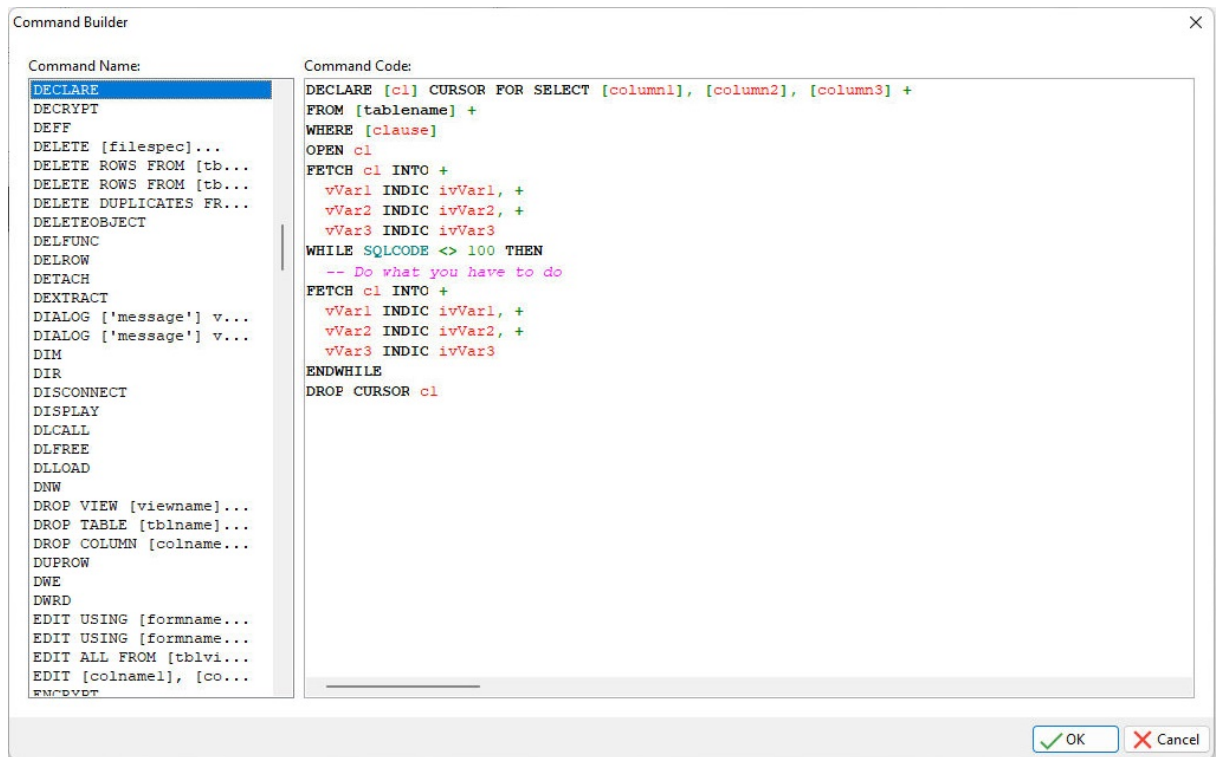
The top of the template list includes three comment templates. Comment templates can be used for the double hyphen "--" comment type, the set of squiggly brackets "{}" for multi-line comments, and the asterisk character preceding a set of parentheses "\*( )". Below is the comment template using squiggly brackets.

```
{ *****
Filename....:
Author.....:
Date Created:
Last Updated:
Purpose.....:

***** }
```

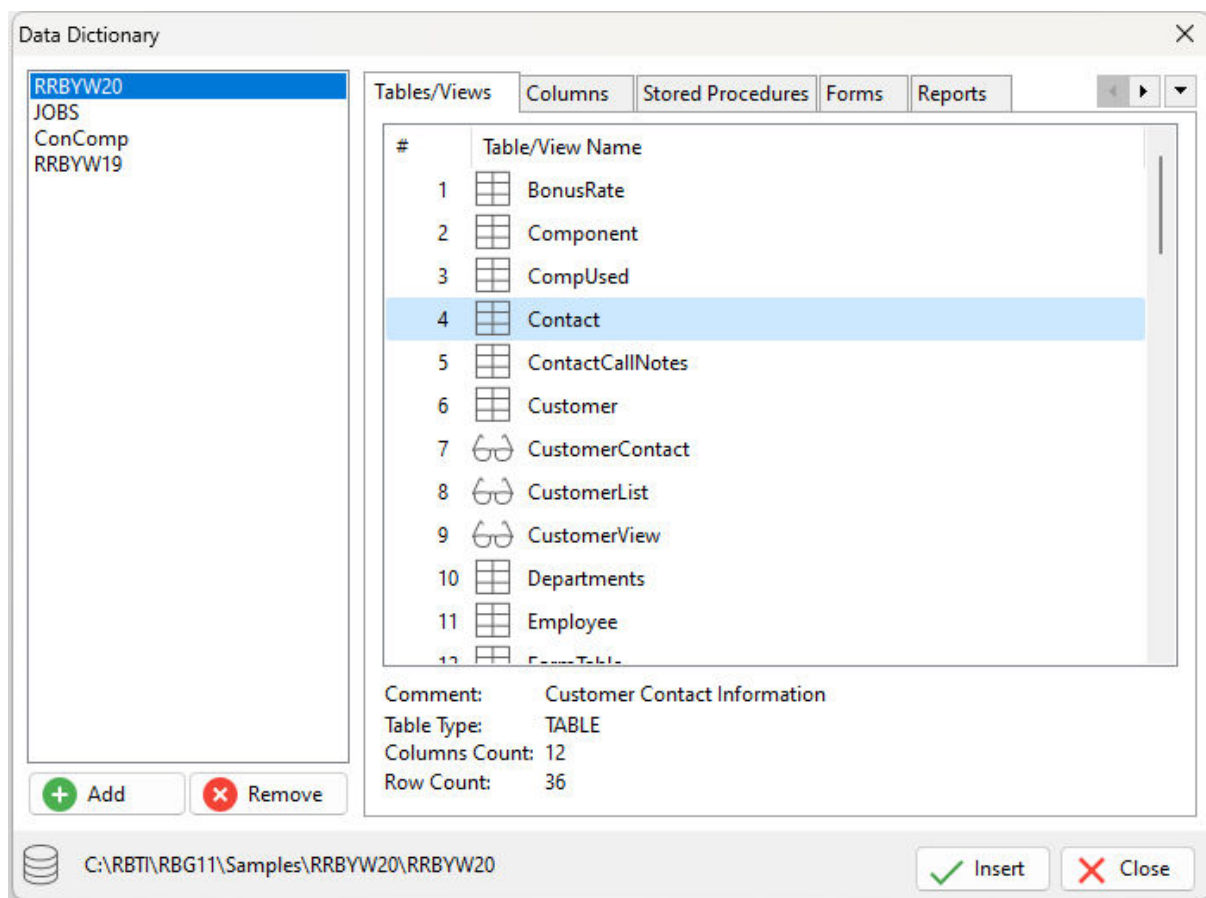
### Build Command at Cursor

Pressing the [Ctrl+F5] keys will display the Command Builder list of commands and functions, sorted alphabetically. Entering a few letters for the desired command or function before pressing [Ctrl+F5] will place the focus upon matching templates. The template displayed in the right panel can be modified and then inserted into the command file/EEP, at the cursor location.



## 2.7 Data Dictionary

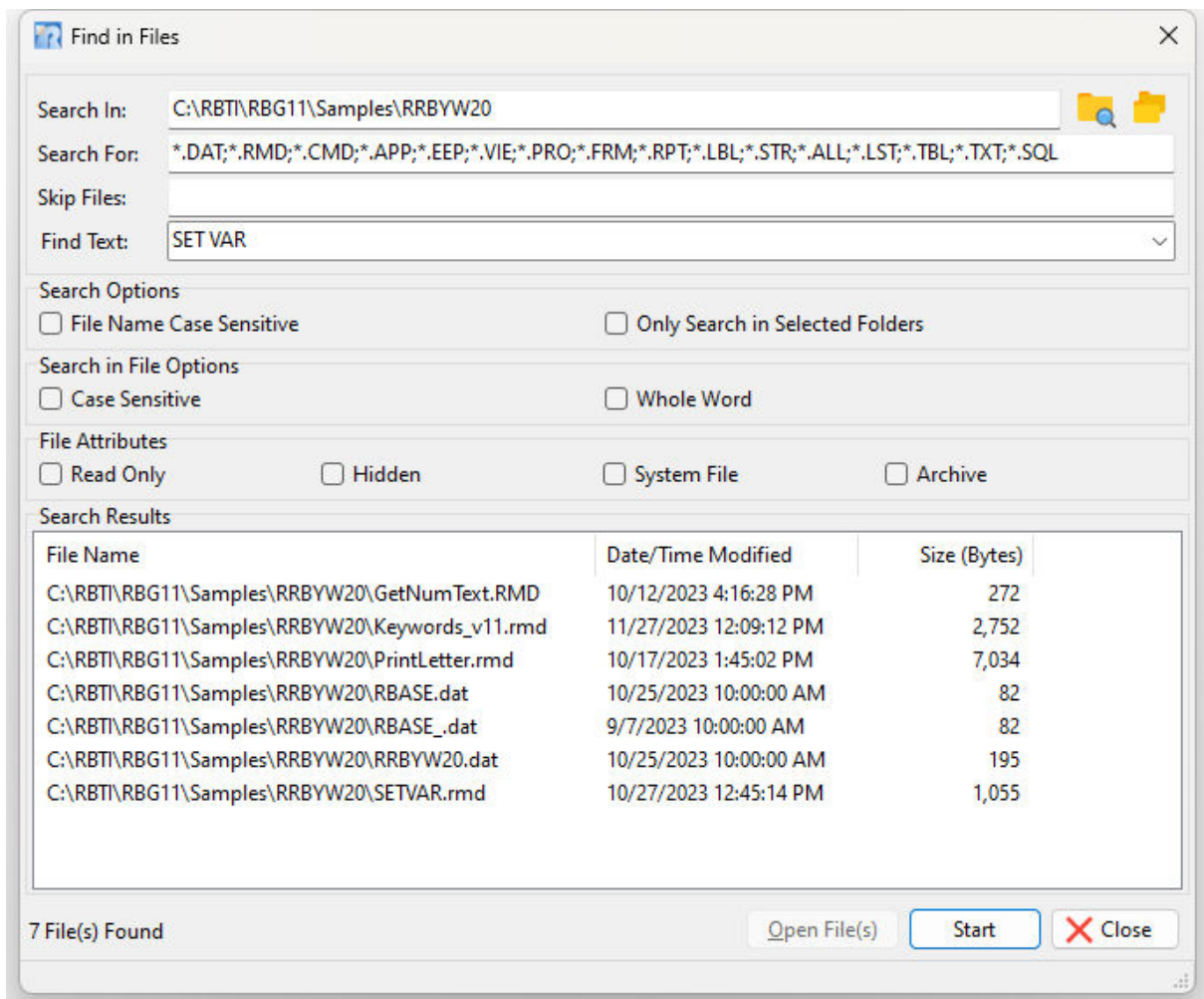
The Data Dictionary is a utility that displays, and allows users to capture, information about the R:BASE environment and/or connected database. It can be launched from anywhere in R:BASE by pressing the [F3] key.



## 2.8 Find in Files

The Find in Files utility allows users to locate command files based upon specific text (commands, keywords, etc.) within the files. Search criteria can be filtered based on file extensions, case sensitivity, multiple directory searches, and file attributes. Using the "Select Multiple Folders..." button, several folders may be specified for the search locations. After the search is complete, the selected files can be opened.

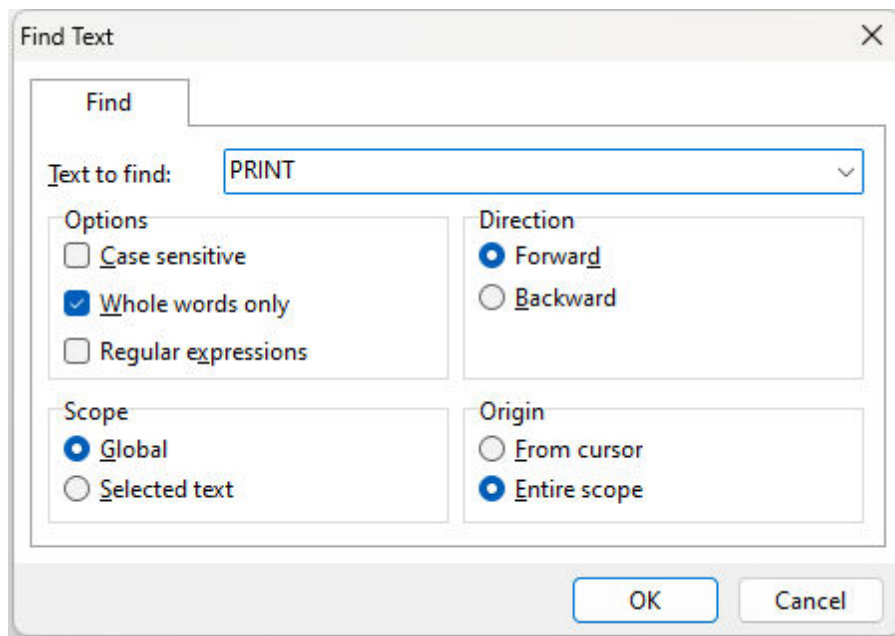




## 2.9 Find Text

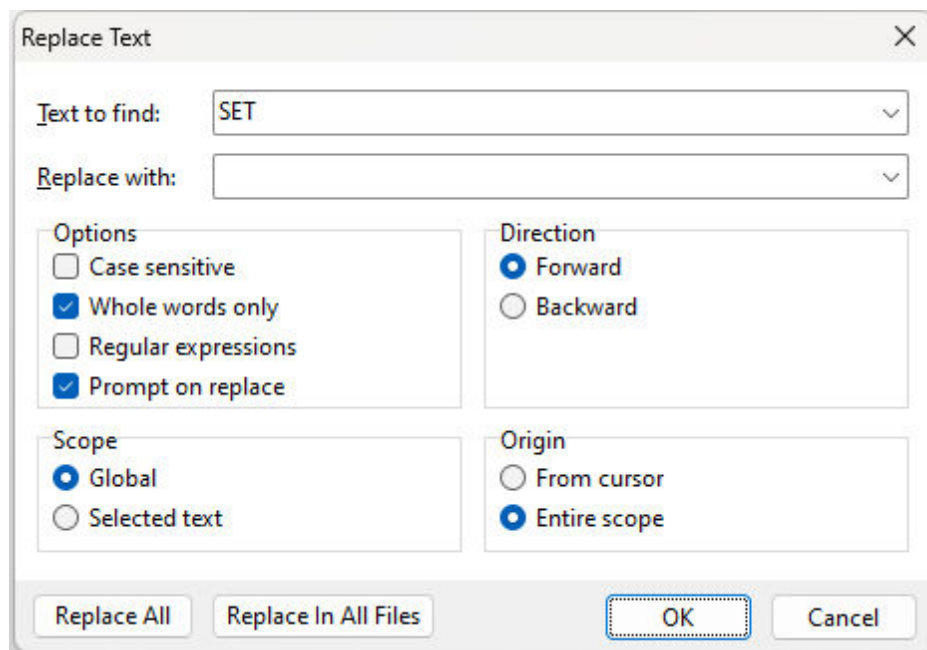
The "Find Text" dialog allows users to locate text in a command file, Custom EEP, Custom Form Action, and also the Trace Debugger. Search criteria can be modified based on case sensitivity, whole word searches, and [regular expressions](#). The search direction, scope, and origin may also be specified.





## 2.10 Replace Text

The "Replace Text" dialog allows users to locate text in a command file, Custom EEP, Custom Form Action, and also the Trace Debugger. Search criteria can be modified based on case sensitivity, whole word searches, [regular expressions](#), and prompt on replace. The search direction, scope, and origin may also be specified.

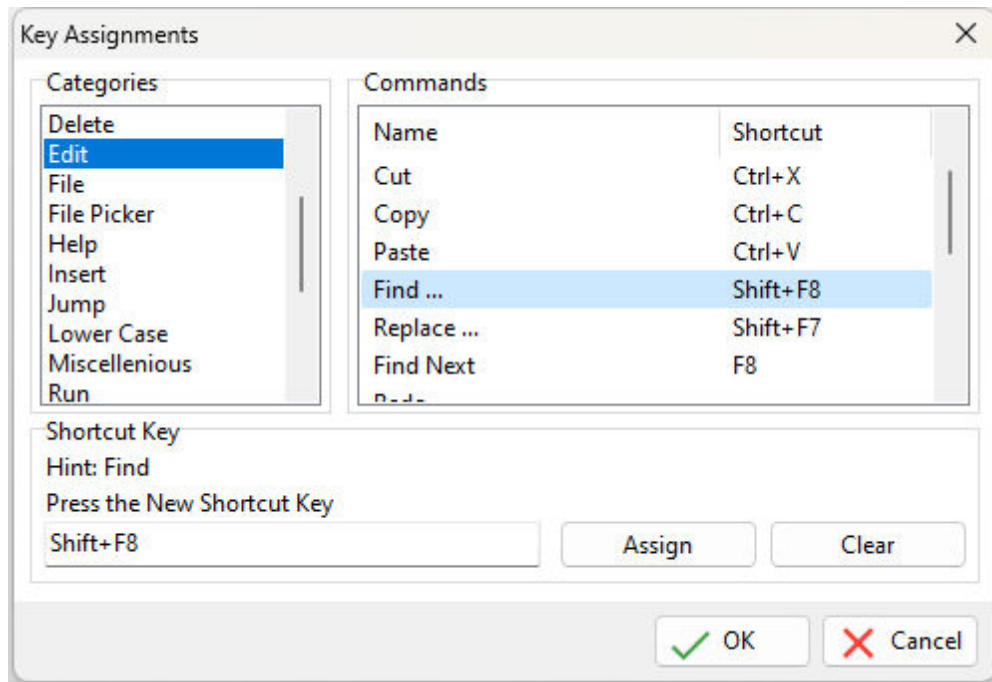


**Replace All** - replaces text for all instances within the current command file. The "Prompt on replace" option may be used to confirm each text update.

**Replace In All Files** - replaces text for all instances within all command files opened as tab within the R:BASE Editor. After Performing [Find in Files](#) searches, the "Replace In All Files" feature is available to replace the text for the searched criteria.

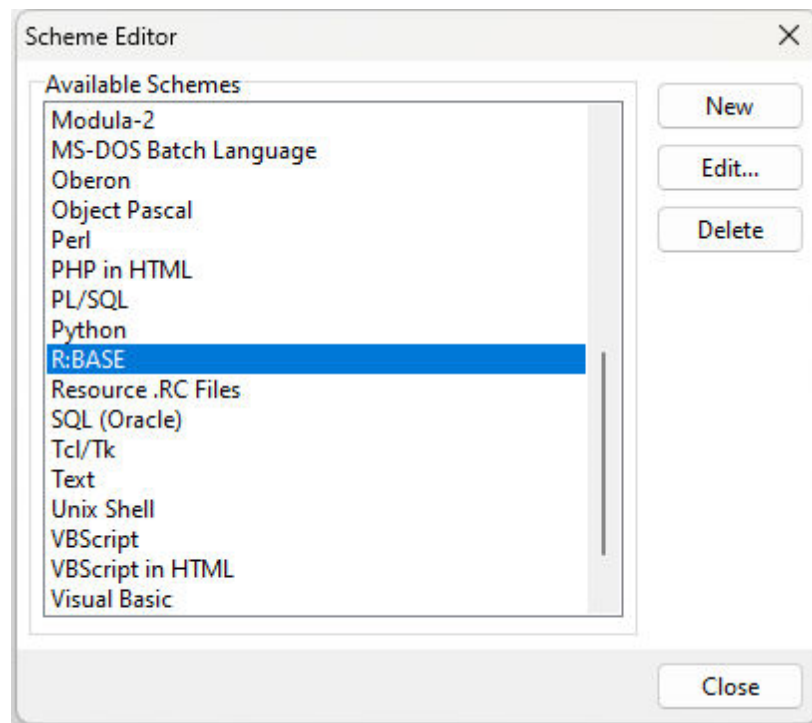
## 2.11 Key Assignments

The Key Assignments option allows the creation of custom key maps and the editing of existing key maps with R:BASE Editor.

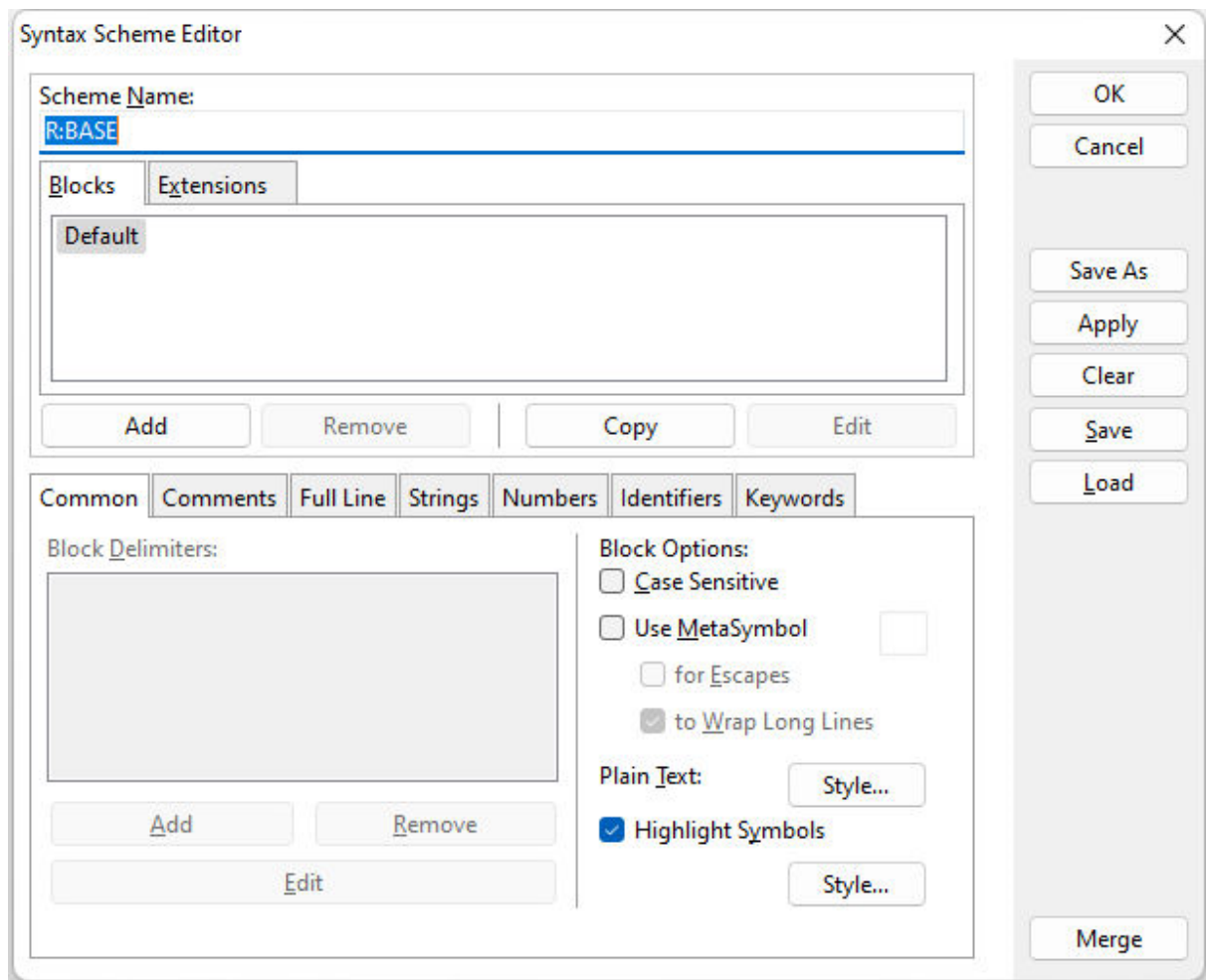


## 2.12 Edit Syntax Scheme

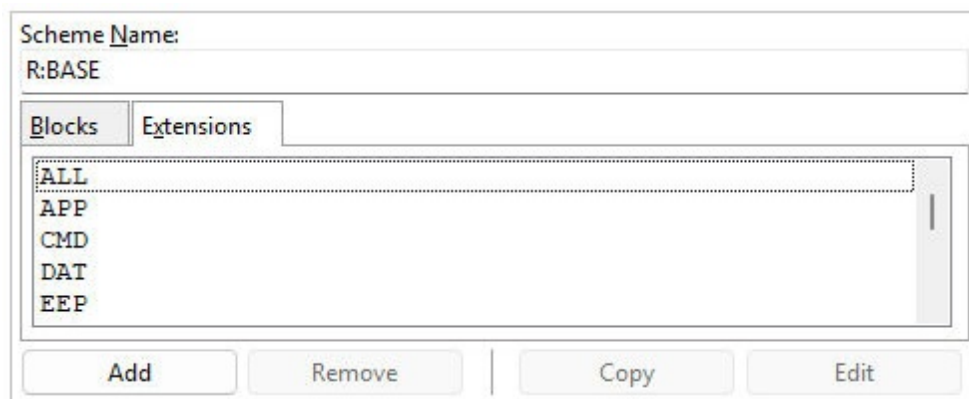
Any of the objects within a command file can hold different background and font color attributes. Syntax highlighting only affects the way your code is displayed in the editor. The programs are still saved as straight ASCII text files. To view the available syntax schemes, select the "Edit Syntax Scheme" button, or select "Options > Edit Syntax Scheme..." from the menu bar.



To change any settings for the R:BASE scheme, select "R:BASE" from the Scheme Editor window, and then the Edit button. The following dialog will appear. The Common, Comments, Full Line, Strings, Numbers, Identifiers, and Keywords tabs are used to assign font color and style attributes based words and characters within the file. To edit any object for the R:BASE syntax scheme, select the tab from the lower panel to view the options available. Selecting the "Edit" or "Style" button within the appropriate tab will display options to edit the color, font, style, background, and size of that scheme object. To load a custom scheme from an existing R:BASE Editor environment, use the "Merge" button to merge the settings into the current scheme.



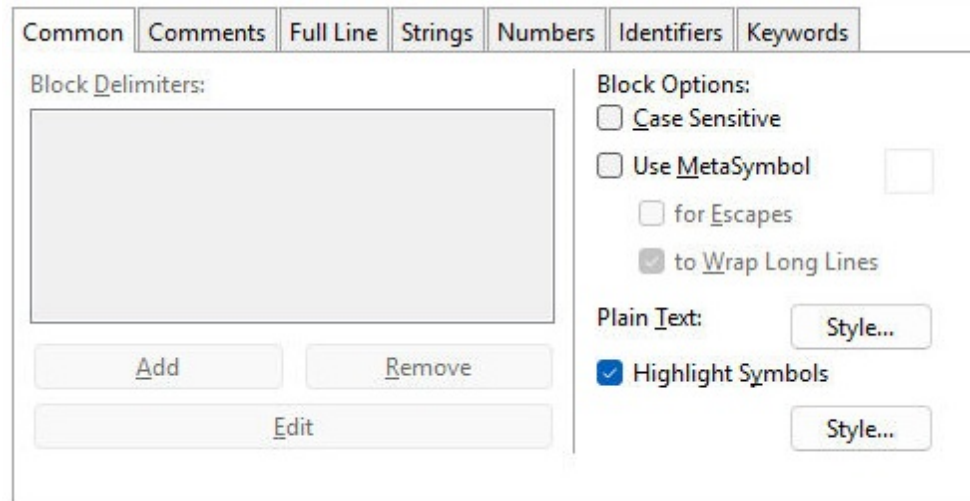
Within the Extensions tab, file extensions associated with the R:BASE scheme may be viewed. When a file with the listed extension is opened in R:BASE Editor the defined syntax highlighting will be assigned to the file contents. The Add, Remove, Copy, and Edit buttons allow for custom extensions to be used.



#### Common

The Common tab is tied to the Blocks tab above, where starting and ending block delimiters may be defined for blocks. Common examples include braces {}, parentheses (), and keywords like *begin* and *end* in certain programming languages. Block delimiters are applicable to other programming languages, but not R:BASE.

*Block Delimiters* - adds, removes, and edits a block delimiter  
*Case Sensitive* - specifies if text block delimiters are case sensitive  
*Use MetaSymbol* - specifies to use a MetaSymbol block delimiter (i.e. braces, brackets, etc.)  
     *for Escapes* - allows the MetaSymbol for escape sequences  
     *to Wrap Long Lines* - allows the MetaSymbol to wrap lines  
*Plain Text / Style* - specifies the text style and font color, style, and name options for plain text  
*Highlight Symbols / Style* - specifies the text style and font color, style, and name options for symbol highlighting



### Comments

Comments in R:BASE are informative statements that are used in files to explain the various routines, constructs, and/or program logic used in the program. The comments help to make the program more readable and understand the code more clearly. Whenever R:BASE encounters a comment, the string is ignored.

R:BASE recognized single-line comments and multi-line comments. Single line comments are written in one line only. They begin with two hyphen characters "--" and extend to the end of the line. R:BASE recognizes the beginning of the comment and ignores everything that follows it. Multi-line comments are the comments enclosed between open and close braces "{" and "}", or the asterisk preceding a set of parentheses "\*()". Everything inside the block that starts with "{" and ends with "}" or starts with "\*(" and ends with ")" is ignored by R:BASE. The multi-line comments can be used on one line, but can extend to multiple lines.

*Highlight Comments* - specifies if comments are highlighted  
*Single-line Comments* - specifies if single-line comments are highlighted, along with options to add, remove, and edit the comment definition  
*Multi-line Comments* - specifies if multi-line comments are highlighted, along with options to add, remove, and edit the comment definition

Common Comments **Full Line** Strings Numbers Identifiers Keywords

☒ Highlight Comments

☒ Single-line Comments

☒ --

☒ Multi-line Comments

☒ { and }

☒ \* ( and )

### Full Line

The Full Line tab allows for "Full-line Comments" with options to add, remove, and edit the comment definition. Full-line comments exist already in R:BASE as double hyphen characters. Other full-line comments are applicable to other programming languages.

*Highlight Directives* - specifies to highlight directives for editors and interpreters

*Prefix / Style* - specifies the directive prefix, and also the text style, font color, style, and name options

Common Comments **Full Line** Strings Numbers Identifiers Keywords

☐ Full-line Comments

☐ Highlight Directives

Prefix:

### Strings

String can include a comment description, caption, title, message, variable using TEXT, NOTE, WIDETEXT, WIDENOTE, alias name, DSN-less connection string, function or procedure with external function declarations (CDECL and STDCALL), etc.

*Highlight Strings* - specifies if strings are highlighted

*Single-line Strings* - specifies if single-line strings are highlighted, along with options to add, remove, and edit the string definition

*Multi-line Strings* - specifies if multi-line strings are highlighted, along with options to add, remove, and edit the string definition

Common Comments Full Line Strings Numbers Identifiers Keywords

☒ Highlight Strings

☒ Single-line Strings ☐ Multi-line Strings

☒ ' and '

☒ " and "

Add Remove Add Remove

Edit Edit

### Numbers

Numbers represent values containing one or more 0-9 characters. Different styles may be applied for integer, float, and hexadecimal numbers.

*Highlight Numbers* - specifies if numbers are highlighted

*Integers* - specifies the text style, font color, style, and name options for integer numbers

*Floats* - specifies the text style, font color, style, and name options for float numbers

*Hexadecimal* - specifies the text style, font color, style, and name options for Hexadecimal numbers

*Special* - allows for the number prefix, suffix, or both prefix & suffix to be defined. The selected radio button determines the options to add, remove, and edit the prefix, suffix, or prefix & suffix definitions.

Common Comments Full Line Strings Numbers Identifiers Keywords

☒ Highlight Numbers ☐ with prefixes

Integers: Style...

Floats: Style...

Hexadecimal: Style...

Special:

☒ with prefix

☐ with suffix

☐ with prefix & suffix

Add Remove

Edit

### Identifiers

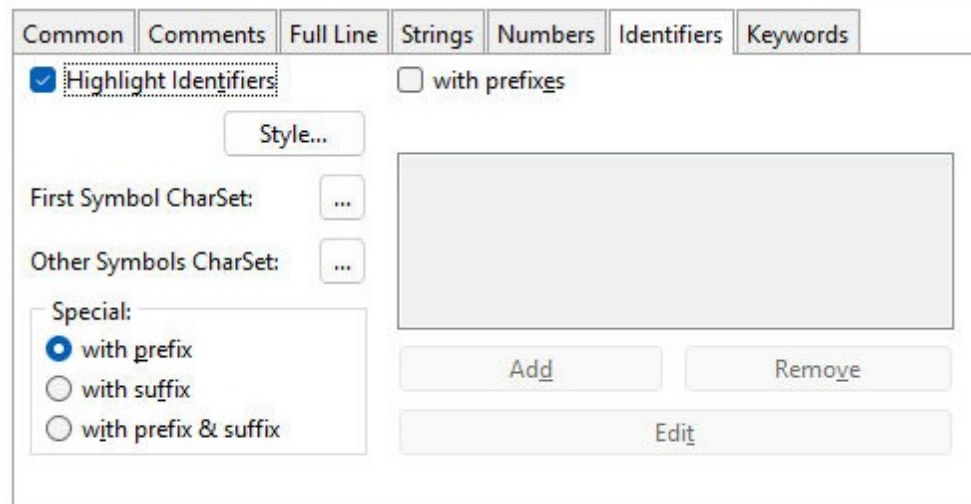
Identifier describes the word used for objects (database, table, column, index, view, stored procedure, form, report, label, correlation, cursor, owner, user, password, alias, arguments, etc.), variables, values for command parameters (plugin, label, component ID, etc.), file names (external form, application, command file, etc.), and folder names.

*Highlight Identifiers / Style* - specifies if identifiers are highlighted, and the ability to define the text style, font color, style, and name options

*First Symbol CharSet* - specifies the character set for the first symbol in an identifier value

*Other Symbols CharSet* - specifies the character set for the other symbols in an identifier value

*Special* - allows for the number prefix, suffix, or both prefix & suffix to be defined. The selected radio button determines the options to add, remove, and edit the prefix, suffix, or prefix & suffix definitions.



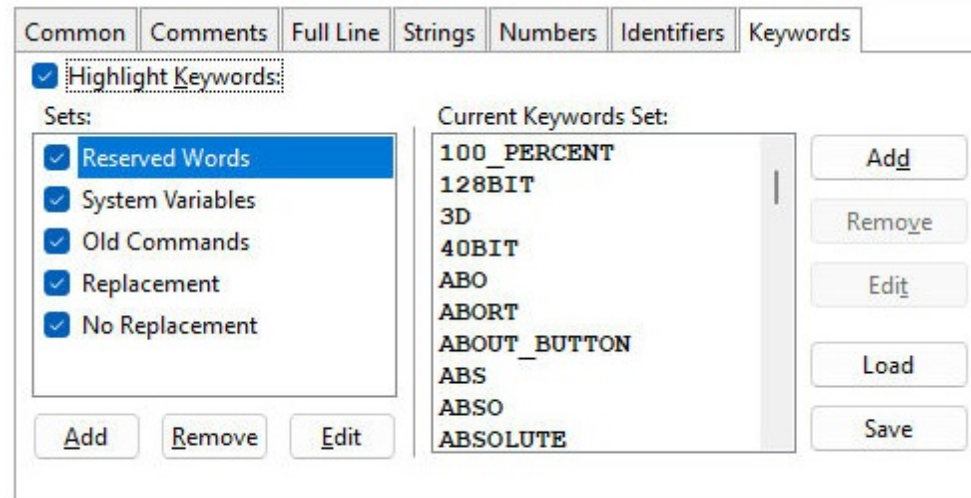
### Keywords

Keywords are specific words which are reserved by R:BASE for its own use. These are mainly commands, functions, data types, parameters for commands and functions, etc. Keywords should not be used to name tables, columns, variables, or any other identifier.

*Highlight Keyword* - specifies if keywords are highlighted

*Sets* - specifies the different keyword sets, which can be added and removed. The Edit button provides the ability to define the text style, font color, style, and name options to specify different highlighting for each keyword set

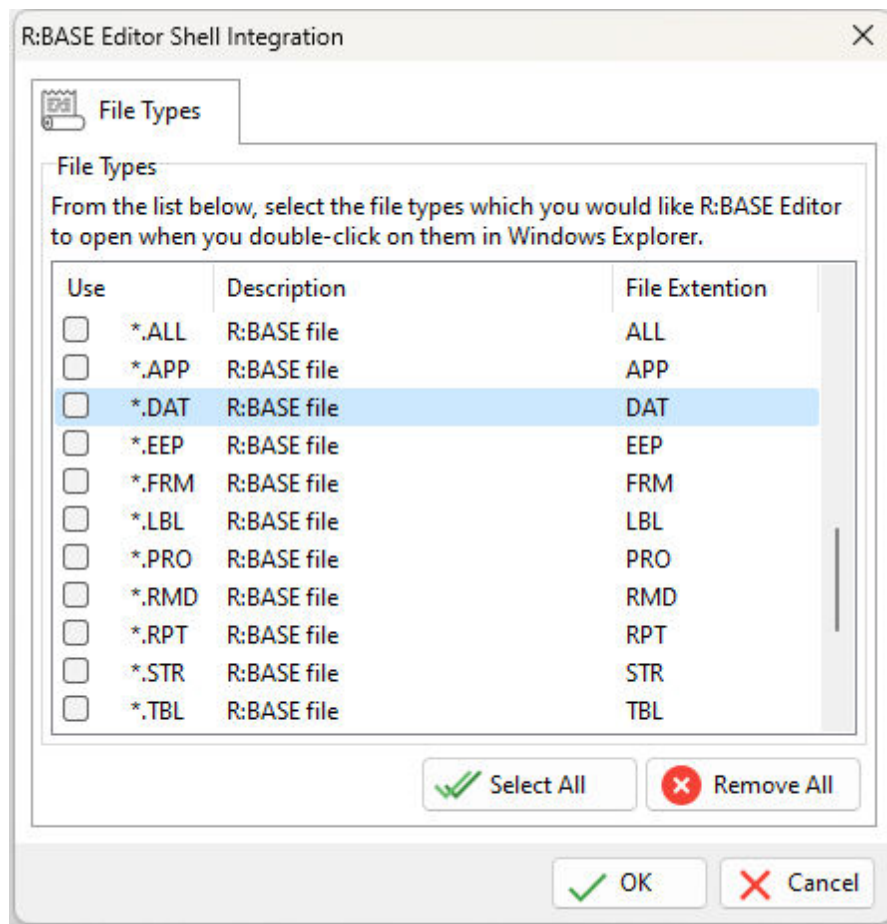
*Current Keywords Set* - provides the set list, and allows for keywords to be added, removed, edited, loaded, and saved



## 2.13 Shell Integration

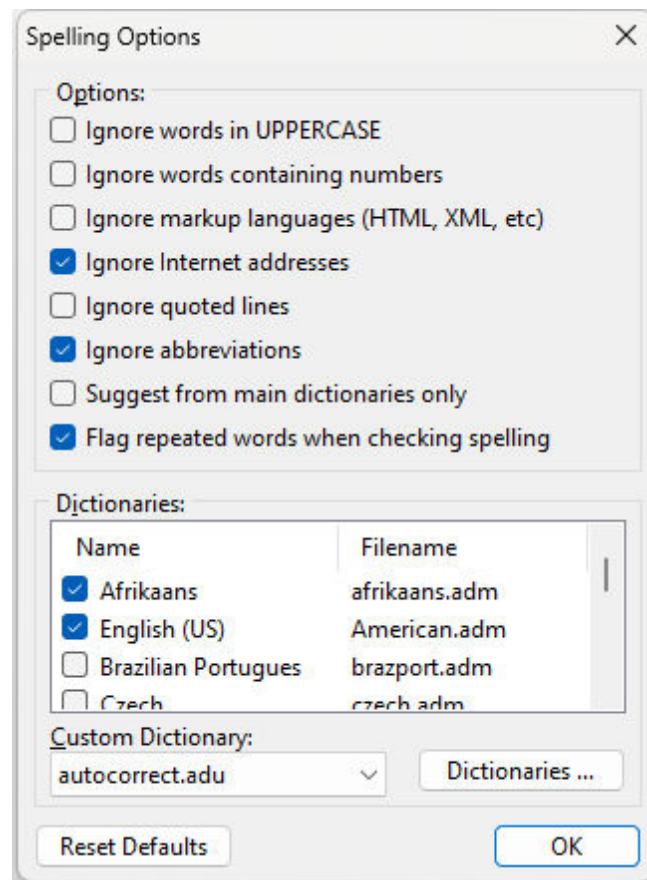
The Shell Integration dialog allows you to associate various file extensions to R:BASE Editor. When you double-click on the file, R:BASE Editor will launch with the command file displayed.





## 2.14 Spell Checker

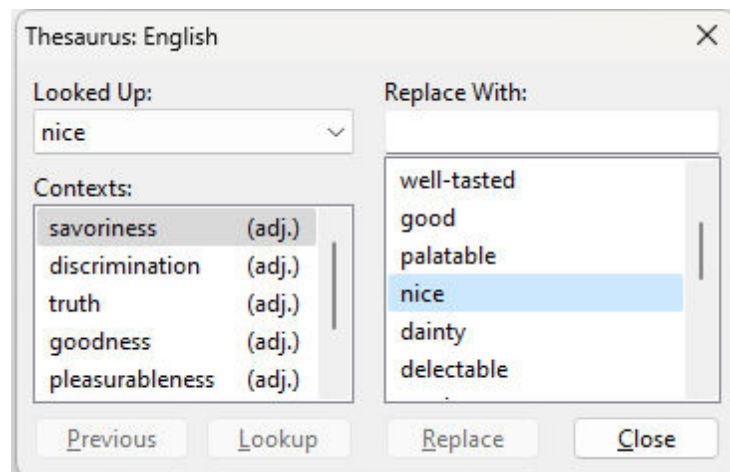
The Spell Checker Setup displays the dialog of the configuration controls specifically related to the spell checker.



## 2.15 Thesaurus

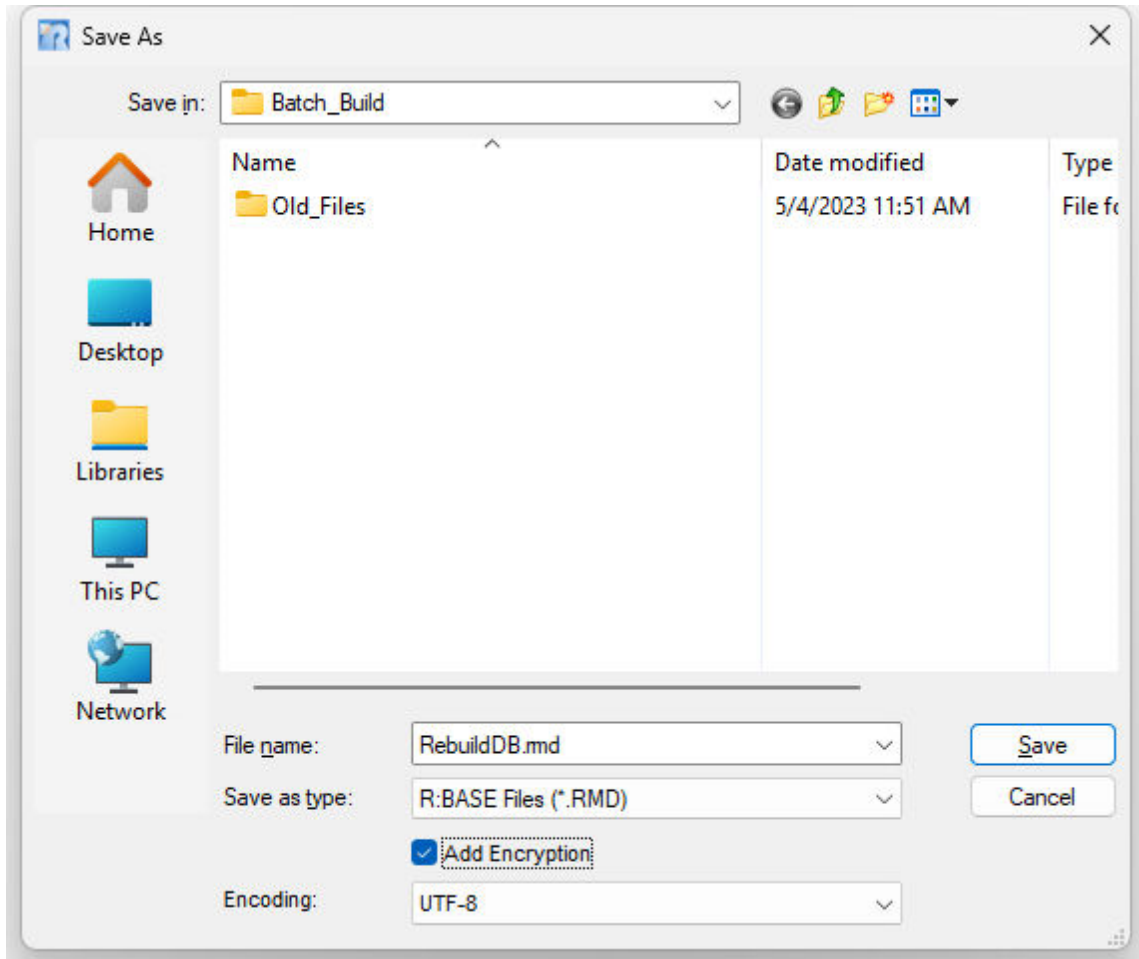
The thesaurus within R:BASE Editor will allow you to search for replacement words for message portions of your R:BASE command files, like PAUSE, DIALOG, and CHOOSE. The thesaurus would also be used for any other text documents opened within R:BASE Editor where the need is to find word synonyms and antonyms.

To launch the thesaurus, select "Run" > "Thesaurus" from the main Menu Bar, or select the thesaurus button on the "Run" toolbar.

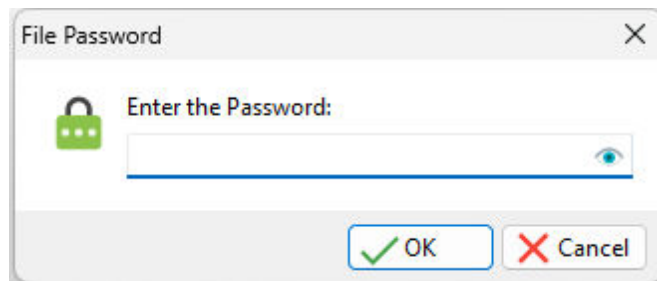


## 2.16 File Encryption

A built-in 512-bit encryption method is supported in the R:BASE Editor for all command files, EEPs, etc., with the File > "Save" or "Save As..." options. The encryption method is used to obscure any file information, making it unreadable without R:BASE and your decryption password.



When you enable the "Add Encryption" check box and select the "Save" button, you will be prompted to enter a 32-character password. The password field includes an "eye" button on the right side. When the "eye" button is clicked down with the mouse, the password is displayed, then hidden again when the mouse button is released.



When running encrypted files within R:BASE, you would issue RUN command followed by file name and password. When opening an encrypted file for editing, you will be prompted for the password.

## 2.17 Using Regular Expressions for Finding Text

Regular expressions consist of patterns that can be used to search for variable forms of text. Without regular expressions, you must know the exact phrase for which to search; regular expressions allow you to find all text that follows a certain pattern. Even if the "Regular expressions" check box in the Find/Replace dialog is on, you can still search for exact phrases. In such a case, you have to remove the regular expression operators from the search criteria.

### Syntax of Regular Expressions for Finding Text

Regular expressions generally consist of some constant text, and some restrictions or wildcards on the rest of the text. If you want to use one of these operators as literal text, you must precede the character with a backslash.

The following is a list of regular expressions used only for searching text. When searching text, you can also use regular expressions that are used for all types of operations.

Character	Meaning
\	Escape character. If you want to use the character '\' itself, you should use '\\'.
^	Line start
\$	Line end
*	Character to the left of the asterisk in the expression should match 0 or more times. For example, 'be*' matches 'b', 'be' and 'bee'.
+	Character to the left of the plus sign in the expression should match 1 or more times. For example, 'be+' matches 'be' and 'bee' but not 'b'.
.	Matches any character.
[ ]	Any of the enclosed characters may match the target character. For example, '[ab]' matches 'a' and 'b'. '[0-9]' matches any digit.
[^ ]	None of the enclosed characters may match the target character. For example, '[^ab]' matches all characters except 'a' and 'b'. '[^0-9]' matches any non-digit character.
-	When used within a character class, represents a character range.
{ }	Create a group. E.g. "{[0-9][0-9]}" is a group that is composed of 2 digits. The search string "{[0-9][0-9]}+" will find the first 8 digits in "123456789" because that is 4 pairs of 2 digits.

### Examples

#### Example 01:

Find a line that starts with "A" and ends with "2":

Search criteria: ^A.\*2\$

^A - Line must start with an A

.\* - Zero or more characters (any character)

2\$ - Line must end with a 2

Search sample and result:

Abc-12 - Found

A2 - Found

1Abc-12 - Not found (line does not start with A)

Abc-123 - Not found (line does not end with 2)

#### Example 02:

Find text that contains the word "Plan," followed by optional space and a 4-digit number:

Search criteria: Plan \*[0-9][0-9][0-9][0-9]

Plan - The literal word 'Plan' must start the phrase

\* - (space character followed by \*) Zero or more space characters

:d:d:d:d - Any four digits

Search sample and result:

My Plan1234 - Found

Plan 1234 - Found  
My Plan 123a - Not found (only 3 digits follow Plan)

**Example 03:**

Find text that contains the characters ' [# ' but is not followed by a 1, 2, 3, 4, or B:

Search criteria: \[#(^1-4B]

- \[ - The literal character [ (must be escaped)
- # - The literal character # (not an operator, does not need to be escaped)
- [^ - Starts a character class, meaning any character but those described in the class
- 1-4B - Range 1-4 (1, 2, 3, 4) and B
- ] - Ends the character class

Search sample and result:

Part [#9778C] - Found  
Lot [#554] - Found  
Part #977 - Not found (# not preceded by [)  
Part [#155A] - Not found (# followed by 1)

**Example 04:**

Find a line that starts with "Detail," followed by one or more spaces, followed by one or more alphanumeric characters, and a Z:

Search criteria: ^Detail+[A-Z0-9]+Z

- ^Detail - Line must start with Detail
- + - (space character followed by +) One or more space characters
- [A-Z0-9]+ - One or more ASCII alphanumeric characters
- Z - The literal character Z

Search sample and result:

Detail 143Z - Found  
Detail AABZ - Found  
Plot A Detail 3Z - Not Found (line does not start with Detail)

**Example 05:**

Find currency values in the form \$#,###.## :

Search criteria: \\$[0-9,]+\.[0-9][0-9]

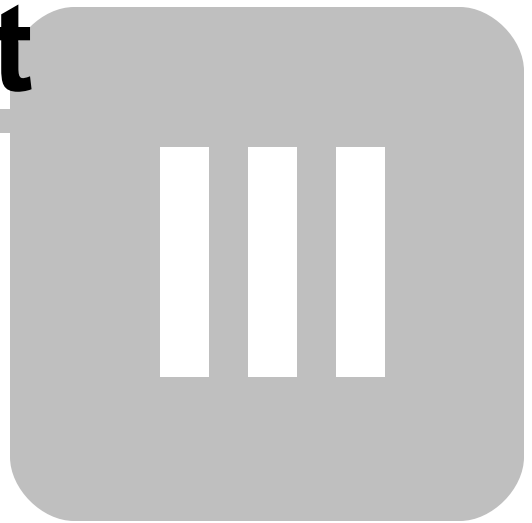
- \\$ - Match must start with a dollar sign
- [0-9,]+ - Matches one or more digits or commas
- \. - Followed by a decimal point
- [0-9][0-9] - Followed by exactly two digits

Search sample and result:

\$12.34 - Found  
\$12,123.00 - Found  
12,435.00 - Not found (does not start with \$)  
\$12 - Not found (does not end with a decimal point and two digits)

# Part

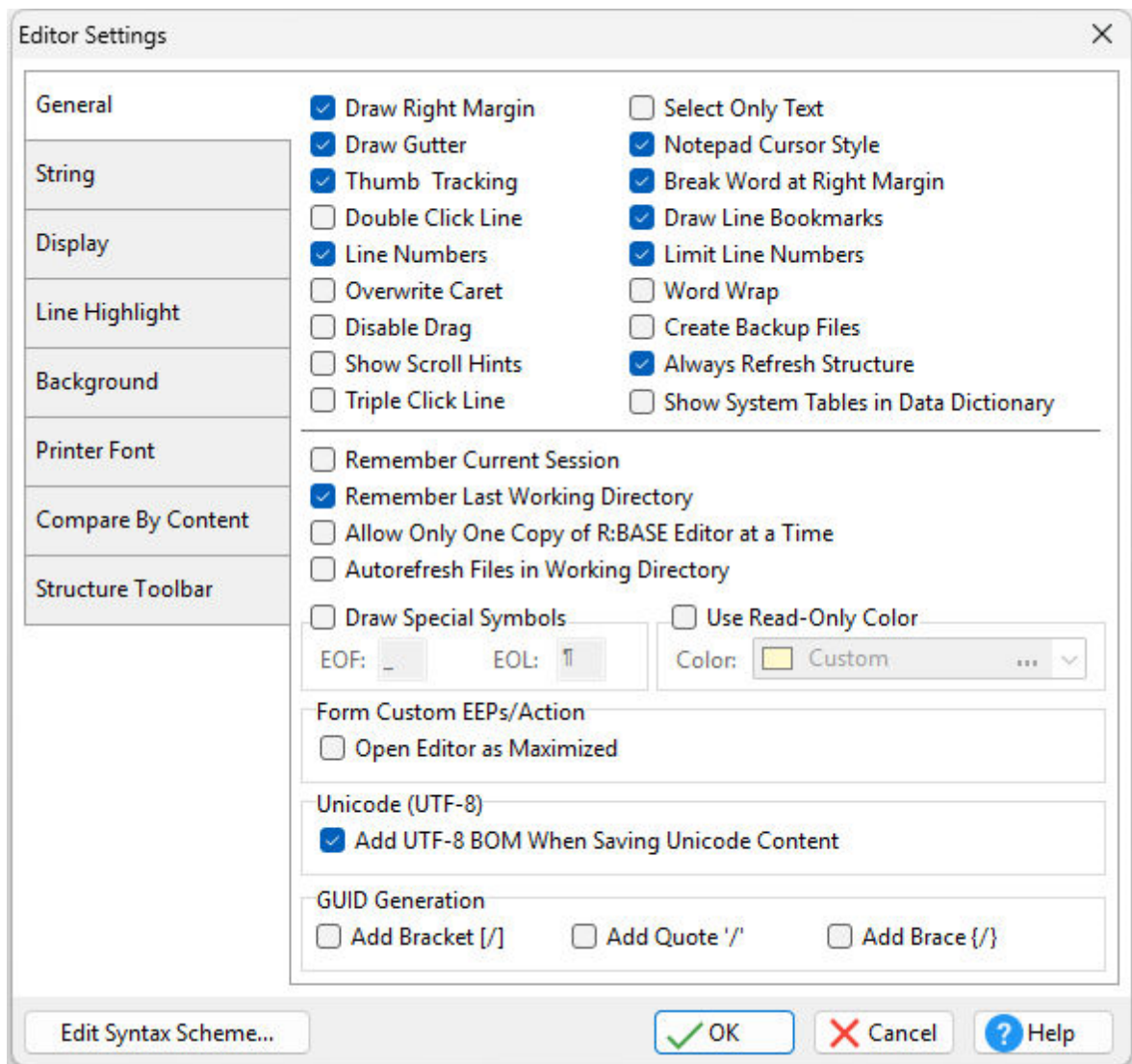
---



## 3 Editor Settings

### 3.1 General

- [Draw Right Margin](#) - enables/disables a right margin in the command file
- [Draw Gutter](#) - enables/disables a gutter on the left margin
- [Thumb Tracking](#) - enables/disables the command file movement before the thumb tab is released while sliding the scroll bar
- [Double Click Line](#) - highlights the line when you double-click any character in the line
- [Line Numbers](#) - enables/disables line numbering in the gutter
- [Overwrite Caret](#) - enables/disables overwrite cursor when the editor is in overwrite mode
- [Disable Drag](#) - disables drag and drop of selected text
- [Show Scroll Hints](#) - displays the line number when dragging the vertical scroll bar
- [Triple Line Click](#) - highlights the line when you triple-click any character in the line
- [Select Only Text](#) - only text is selected when additional lines selected
- [Notepad Cursor Style](#) - allows cursor to not stay on first character in line when left arrow is pressed
- [Break Word at Right Margin](#) - enables/disables word break at the right margin when word wrap is on
- [Draw Line Bookmarks](#) - enables/disables an arrow placement at the cursor location of an added bookmark
- [Limit Line Numbers](#) - enables/disables limitation line numbers to the length of a command file
- [Word Wrap](#) - enables/disables wrapping text
- [Create Backup Files](#) - enables/disables the creation of a backup file once a file is saved (file = CUST.APP backup = CUST.~APP , file = RESTORE.RMD backup = RESTORE.~RMD)
- [Always Refresh Structure](#) - refreshes the Structure tool bar when switching between open files and when the file is saved
- [Show System Tables in Data Dictionary](#) - enables/disables the display of database system table when opening the Data Dictionary
- [Remember Current Session](#) - restores a group of commands which were open in the prior session
- [Remember Last Working Directory](#) - remembers the last working directory, regardless of the "Start in" path when using a desktop shortcut
- [Allow Only One Copy of R:BASE Editor at a Time](#) - allows and prevents multiple sessions of R:BASE Editor
- [Autorefresh Files in Working Directory](#) - allows auto-refreshing of the current working directory
- [Draw Special Symbols](#) - allows the display of special symbols for the end of a file (EOF) and the end of a line (EOL)
- [Use Read-Only Color](#) - allows the display of a color background for read-only lines
- [Unicode \(UTF-8\)](#) - specifies whether to add a UTF-8 BOM to a saved file
- [GUID Generation](#) - specifies the GUI value format when selecting the [Insert GUID](#) menu option

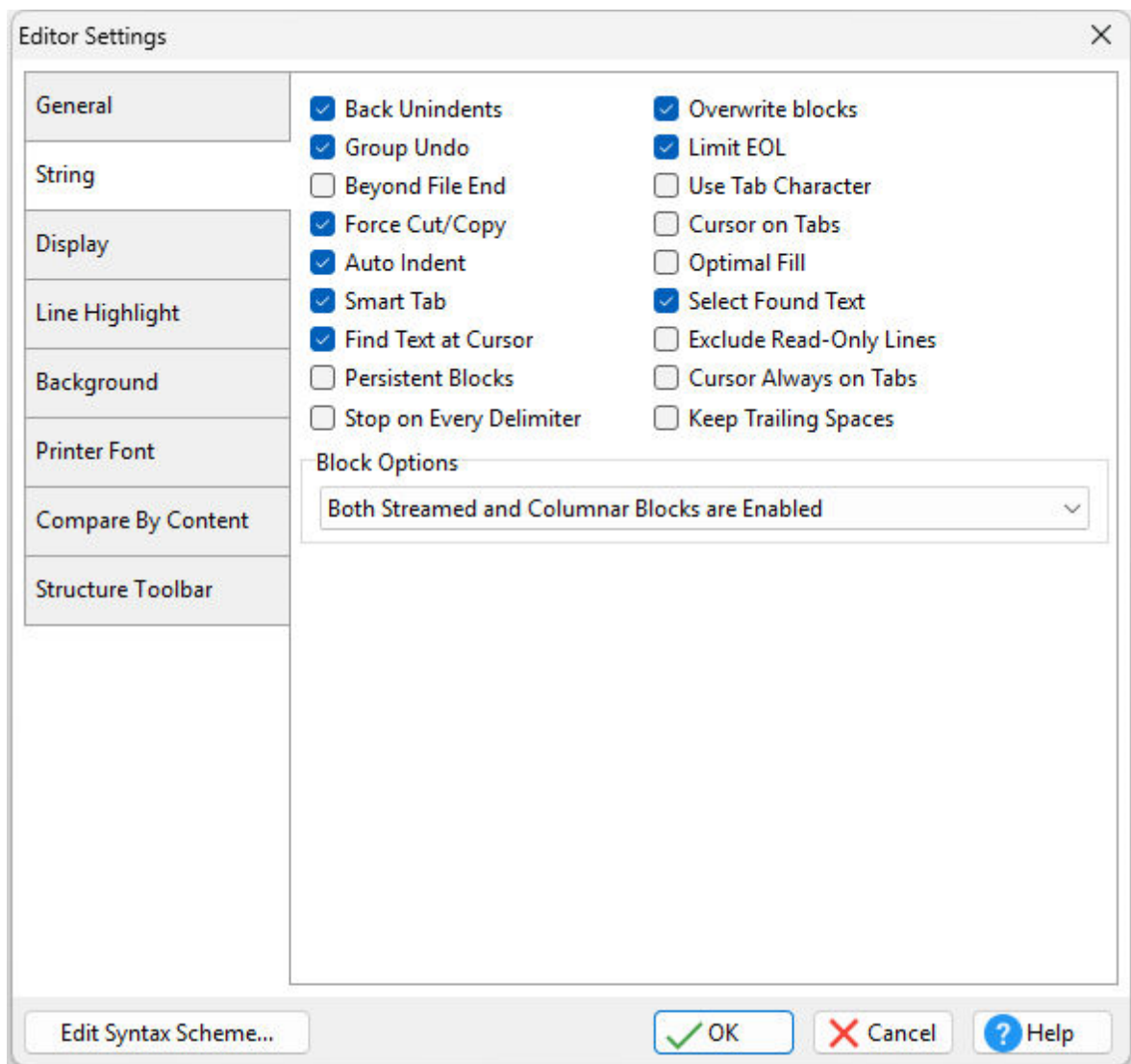


## 3.2 String

- **Back Unindents** - enables/disables alignment of the cursor with the previous indentation level from the preceding line when you press the BACKSPACE key
- **Group Undo** - enables/disables the undoes of your last editing command as well as any subsequent editing commands of the same type
- **Beyond File End** - enables/disables position of the cursor beyond the last character
- **Force Cut/Copy** - enables/disables cut and copy, even when there is no text selected
- **Auto Indent** - enables/disables the cursor to be placed under the first non-white space character in the preceding line when you press the ENTER key
- **Smart Tab** - enables/disables the cursor to be placed under the first non-white space character in the preceding line when you press the TAB key
- **Find Text at Cursor** - enables/disables placement of the text at the cursor into the Text To Find list box in the Find Text dialog box
- **Persistent Blocks** - enables/disables to keep a marked block (selected text) selected even when the cursor is moved
- **[Ctrl + Arrow] Stops on Every Delimiter** - Enables/disables cursor navigation mimicking the R> cursor, where pressing [Ctrl] + [Arrow] Keys will move the cursor to the far left character for every word or combined character string.



- **Overwrite Blocks** - enables/disables to replace a marked block (selected text) with whatever is typed next
- **Limit EOL** - enables/disables the cursor from being positioned past the end of a line
- **Use Tab Character** - enables/disables insert of the Tab character into the text when the user presses Tab button
- **Cursor on Tabs** - enables/disables the caret to be moved by tabs
- **Optional Fill** - enables/disables the beginning of every auto-indented line with the minimum number of characters possible, using tabs and spaces as necessary
- **Select Found Text** - enables/disables to select the text that was found in a Find operation
- **Exclude Read-Only Lines** - enables/disables excluding read-only text
- **Cursor Always on Tabs** - enables/disables pressing right and left arrows will move the cursor on tabs
- **Keep Trailing Spaces** - keeps/removes trailing spaces at the end of code lines
- **Block Options** - options for block selection in command files



### 3.3 Display

#### Gutter

- **Width (Screen Pixels)** - options for width of gutter (0 will base the gutter width on file size)
- **Color** - options for the gutter color
- **Line Number Color** - options for color of numbers in the gutter

- [Line Separator](#) - enables/disables a separator between number lines of code with color and style
- [Right Margin \(Characters\)](#) - options for placement of right margin
- [Tab Stops](#) - stop points for tabbing across a line
- [Block Indent Step Size](#) - number of spaces that a command block will indent using [Ctrl]+[K]+[I]; outdent using [Ctrl]+[K]+[U]

#### Default Font

- [Font Name](#) - specifies the font name
- [Font Size](#) - specifies the font size
- [Enforce Default Font Size](#) - specifies to enforce the font name and size for all word types (comments, strings, numbers, identifiers, keywords)

[Edit Syntax Scheme...](#) - displays the Scheme Editor to alter the syntax highlighting that affects the way code is displayed in the editor

Editor Settings

**General**

Gutter

Width (Screen Pixels): 38

Color: Custom

Line Number Color: White

☐ Line Separator

Color: Gray

Style: Dots

Right Margin (Characters): 80

Tab Stops: 9,17

Block Indent Step Size: 1

**Default Font**

Font Name: Courier New

Font Size: 10

Enforce Default Font Size ☐

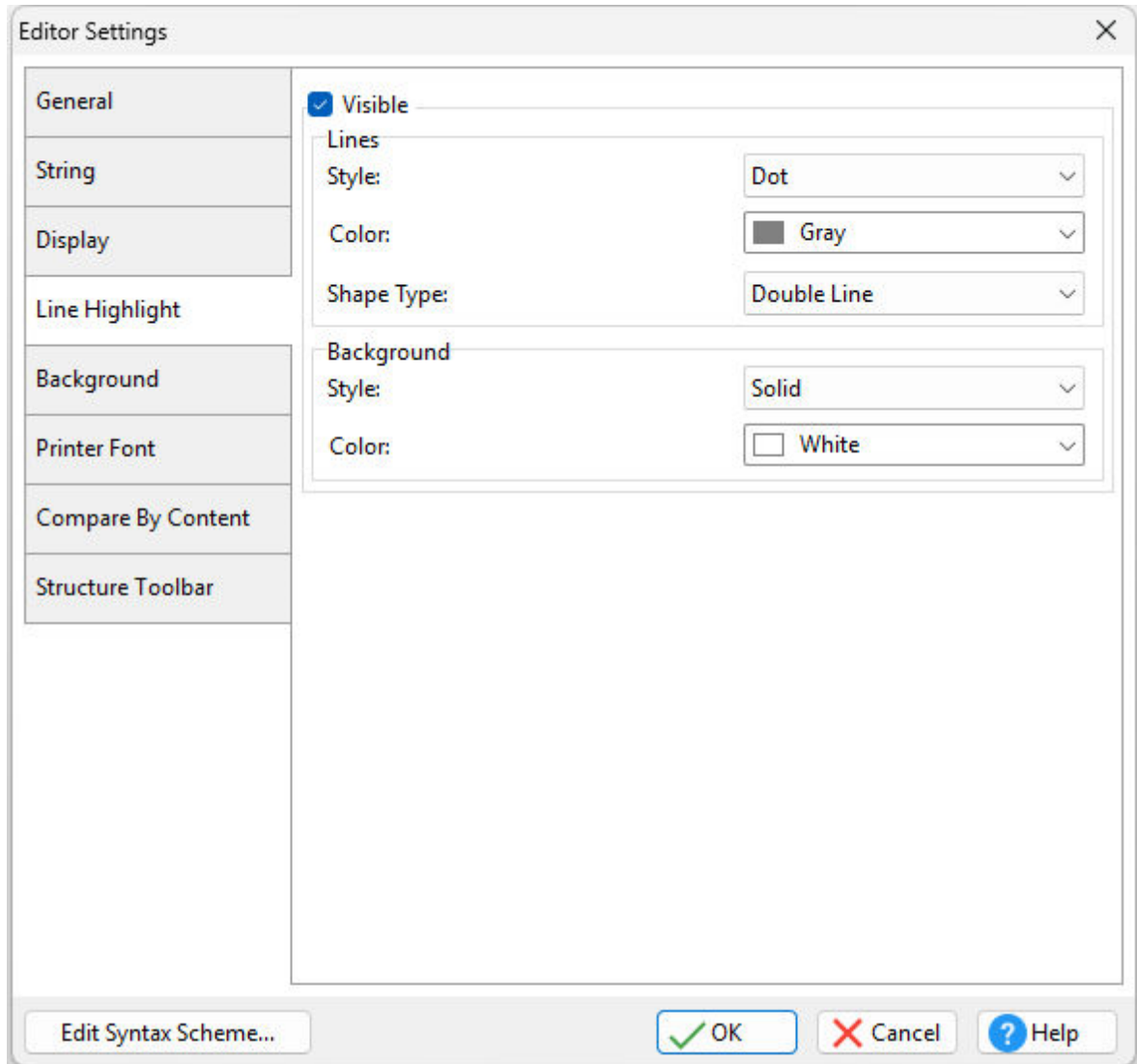
**Default Syntax Scheme**

Edit Syntax Scheme...

OK Cancel Help

## 3.4 Line Highlight

These settings will highlight the current line with a style, color, and background where the cursor is located.

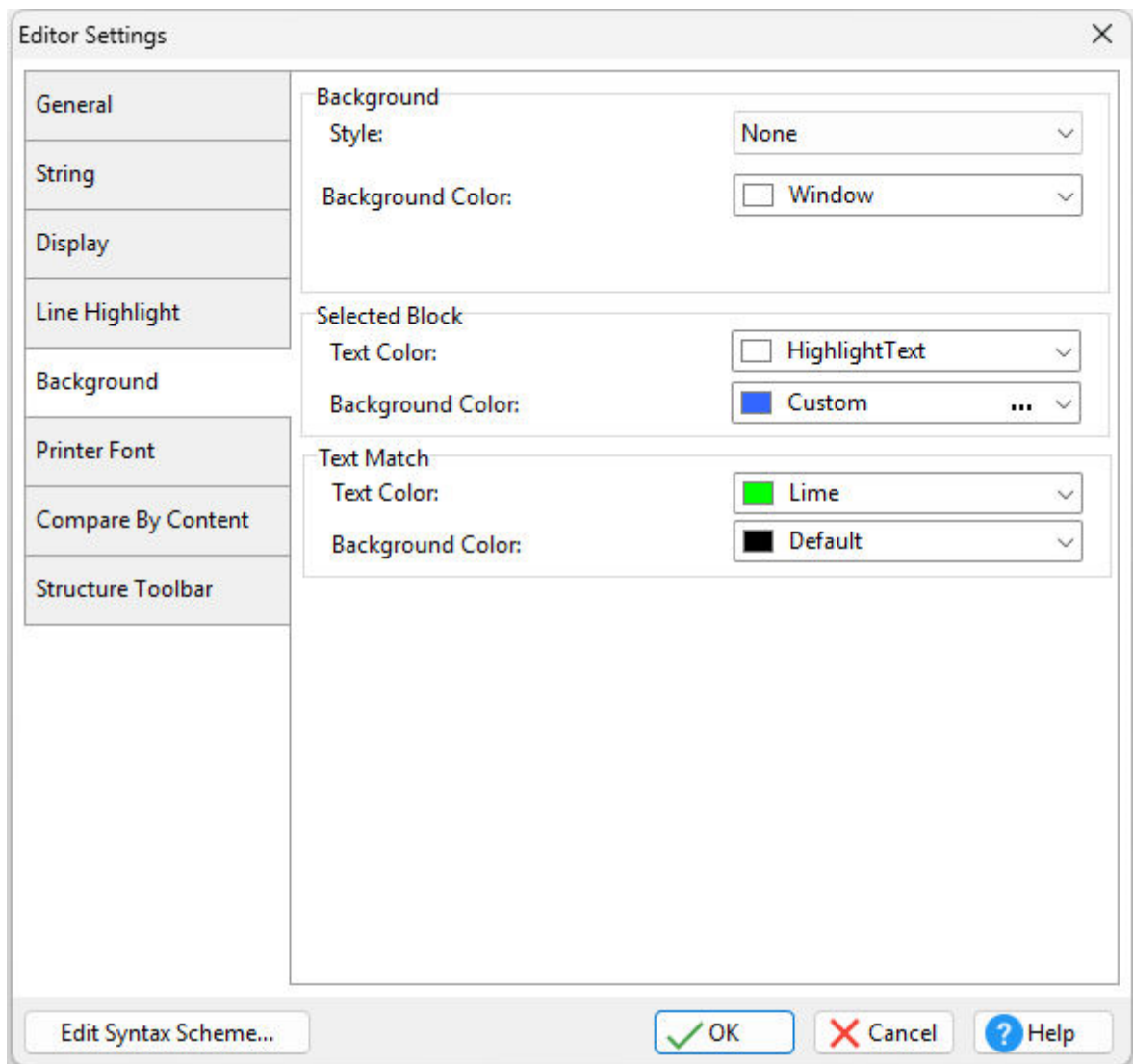


## 3.5 Background

*Background* - will alter the background color of the applied to the Editor window, the Structure tool bar, the File Picker tool bar, and the Error Messages tool bar.

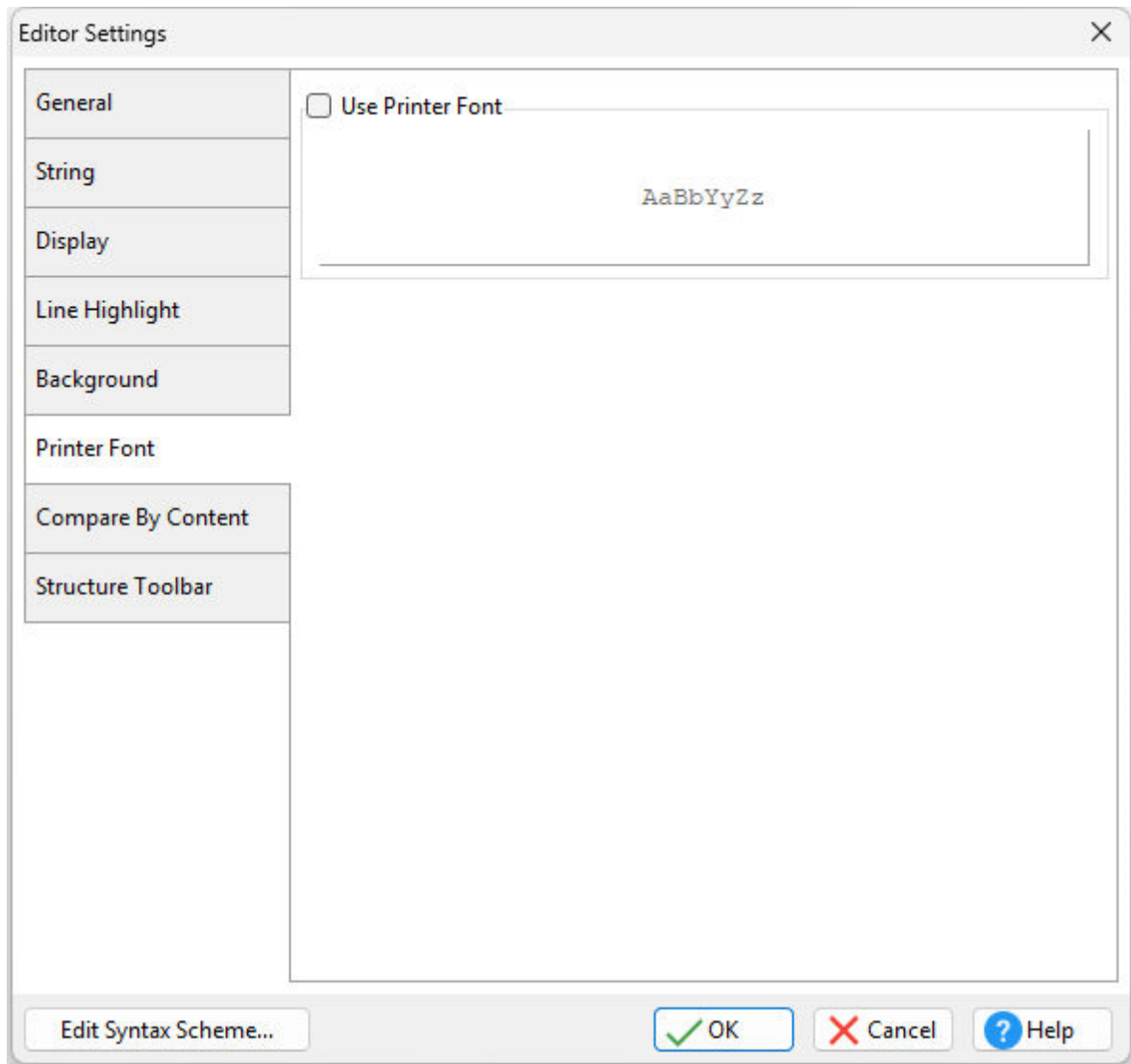
*Selected Block* - will alter the text and background color of the cursor selected text

*Text Match* - will alter the text and background color of the text located in the Find Text utility



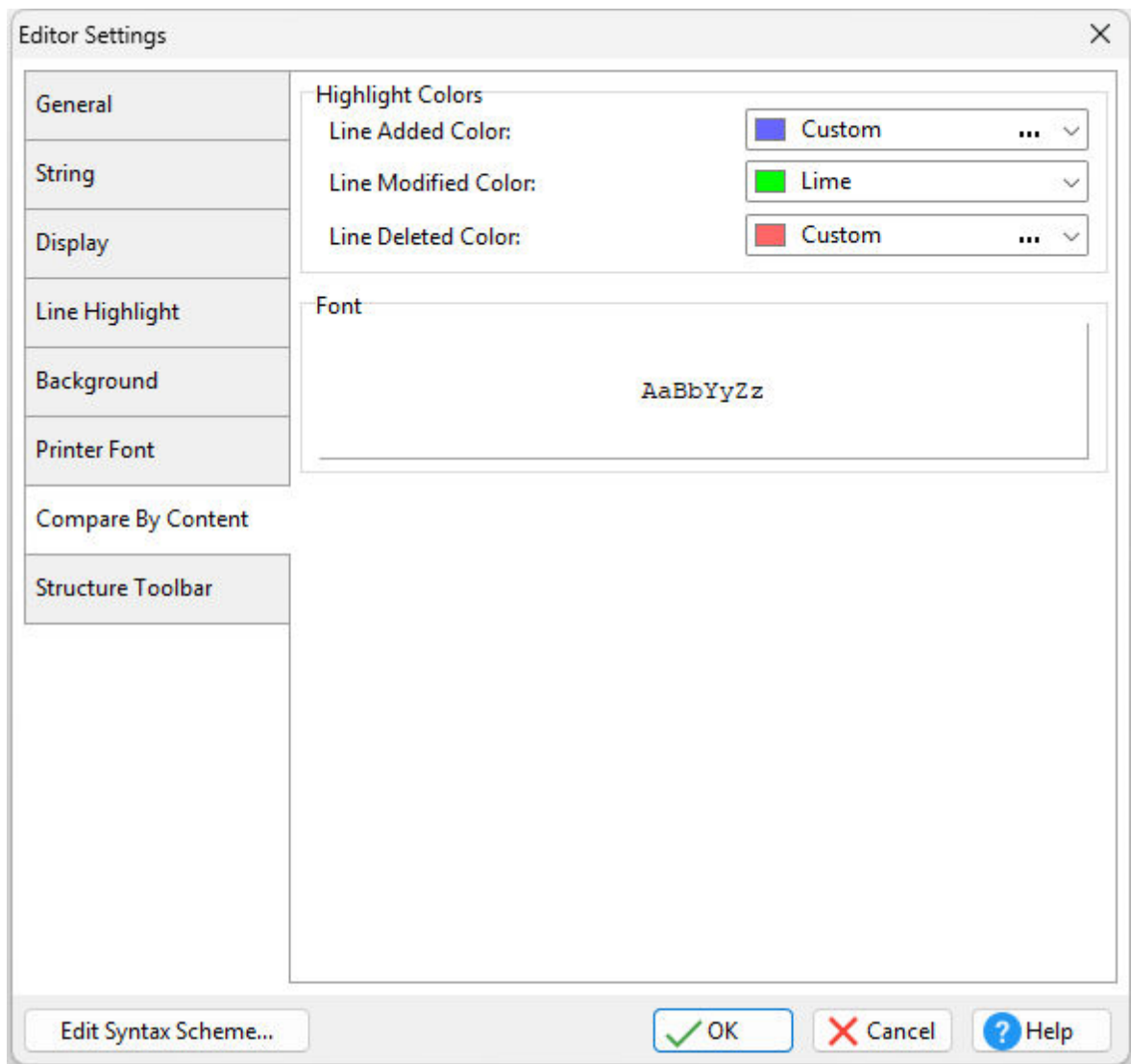
## 3.6 Printer Font

Printer output can be forced to print any of the available fonts.



## 3.7 Compare by Content

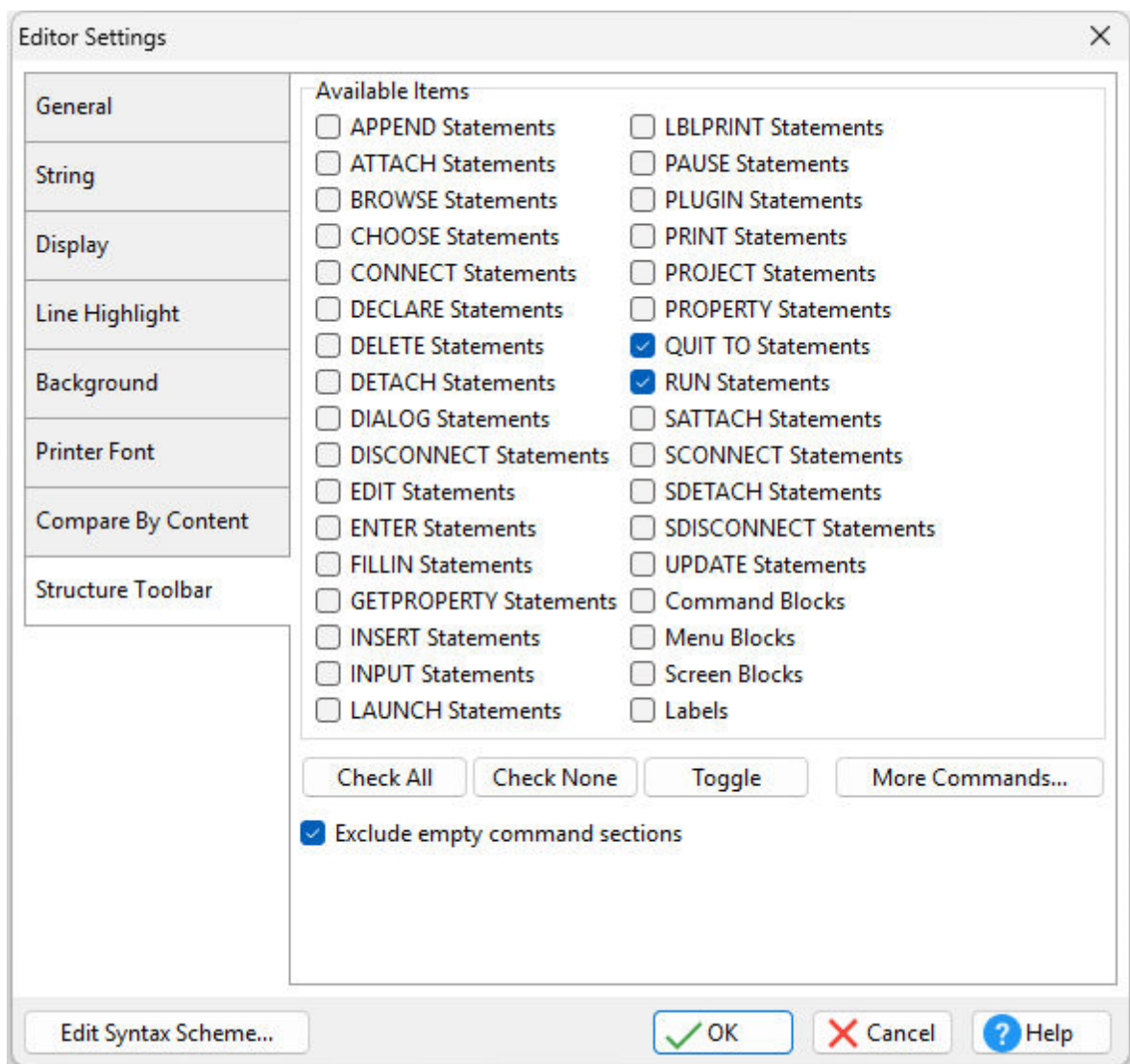
The colors which are displayed when comparing two commands file can be changed to any of the available colors. The font displayed in the [Compare by Content](#) Toolbar can also be changed.



### 3.8 Structure Toolbar

The [Structure Toolbar](#) displays command file blocks and specific statements. Items can be removed from being listed in the toolbar by unselecting the available check boxes.

[Exclude empty command sections](#) - excludes commands from the toolbar and output, when not listed in the command file



**Part**

**IV**



## 4 R:Style

R:Style provides structural error checking, and spelling assistance to help debug your code prior to running it, or help you find errors quickly if a new piece of code drops you to the R> prompt. R:Style catches many errors of coding style which only occasionally cause an application to fail.

R:Style will tell you where in the file to look, and what error to look for! R:Style is not a full syntax checker, but it is an excellent structure checker, with some syntax checking thrown in as a bonus.

If your first step after modifying a command file is to R:Style it, many errors will be caught before you run the code. Why wait for your code to bomb? Why go back to your code and stare at it to see if you can find the error? TRACE is really helpful but some of the errors R:Style traps are ones that cause trace to give up and abort. The few seconds it takes to R:Style the file from within R:BASE Editor will find many of your errors right away, making you MUCH more efficient.

R:Style gives you powerful [tools](#) to change all occurrences of a word with a smart algorithm which knows how to isolate words based on their context. This is really useful when doing conversions.

R:Style automates the process of unifying the style of your command files, while at the same time giving you a great deal of flexibility in matching your current style. This frees you to write code without worrying about the case of the words you are typing.

R:Style allows you to quickly re-style someone else's code to your style so you can read it more easily. This is valuable because of the great diversity of coding styles among R:BASE programmers.

If you are a consultant, you often have the job of cleaning up someone else's code. R:Style makes that code easier to read because it will be in the style you are used to.

In a shop where more than one person is coding, everyone's code can be quickly cased and indented to one style. When the existing code is consistent it is amazing how fast everybody learns to use the standard or, at least, to use this program to style their code.

You can select from many different styles of [indenting](#) and line wrapping, so you are sure to find one close to what you are used to. R:Style's consistency will make you wonder how you managed without it.

R:Style gives you the ability to switch styles if you find one that you like better. All your old code can be restyled in VERY short order. If you don't like the new style, you can always go back.

The automatic word casing and [indenting](#) eliminates having to correct these details while coding so that part of your work can proceed faster. Custom word casing and reporting of unknown words helps you find spelling errors in your code.

Even better, since R:Style looks at ALL the code in your files, it will find some of the errors that may be lurking in the code undetected which is most useful!

### **WARNING**

**ALWAYS BACKUP YOUR FILES** before handing them over to a utility like R:Style. This will save you a lot of grief if anything should go wrong.

### **SECURITY**

R:Style avoids processing quoted text, comments, \$SCREEN or \$MENU blocks, and data in a LOAD block. The built-in keyword expansion tables are carefully crafted to avoid breaking your code. R:Style recognizes SET QUOTES and processes your files properly even if you change the setting in mid-stream. The quotes content of each file is examined and processing is aborted if the majority of quotes contained in the file do not match the character specified in the configuration. Extensions of the files you specified are also checked and an abort occurs if any are in a list of dangerous extensions such as .EXE, .RBF, etc.

In any case, BACKUP YOUR FILES, TEST THE RESULTS.

Obviously, if you process a file twice, the benefit of the backup file is lost. You are responsible to protect yourself by making backups of your files before processing them. We cannot be responsible beyond my own personal desire to provide a bullet-proof product. You have a responsibility to protect yourself. BACKUP YOUR FILES! It is your responsibility to test your code after processing with R:Style to verify

that it still works properly. In most cases, the alterations made by R:Style will enhance your code without breaking it. If your code does break it usually is because you used an abbreviation of a reserved word for a variable or column name and R:Style expanded it to the full word. TRACE will find these problems quickly if they exist and your program will be more bullet-proof after it is corrected. These problem names are easy to see because they will be upper cased.

## 4.1 Features

R:Style Featured Highlights:

- Checks illegal or ambiguous commands
- Checks missing, unmatched, or incorrect quotes
- Checks unmatched parenthesis
- Finds missing RETURNS & ENDS, adds them
- Finds missing or out of place structures, IF, THEN, ELSE, ENDIF, SWITCH, ENDSW, WHILE, THEN, ENDWHILE, BREAK
- Duplicate block names in APP files
- Checks illegal use of command words in code or block names
- Detects duplicate LABEL names
- Checks ill advised RETURN from within WHILE or SWITCH
- Checks illegal dotted variables
- Checks dash comments extending beyond column 79 (breaks code lock)
- Support for finding misspelled or illegal words
- Integration of RStyle\* .PRE, .EXP, .CAS, .VAR, .NEW keyword formatting files into a built-in control panel
- Mismatched lines (= instead of +)
- Local or global configurations
- NO file size limitation
- Improved syntax checking for the SELECT command
- Saves user preferences
- Added compatibility for comments using left and right brace characters. {}
- Fail-safe R:Style configuration file lookup
- Read-only mode available for those situations where you are especially paranoid!
- Cleans up excess EOF markers created by some editors

### Optional Error Checking:

- Optional automatic (and configurable) indenting of structures:

```
IF
  statements...
ELSE
  statements...
ENDIF
```

```
WHILE
  statements...
ENDWHILE
```

```
SWITCH
  CASE -- optional switch to line up CASE with SWITCH
    statements...
BREAK
DEFAULT
  statements...
ENDSW
```

Many other styles supported.

- Labels can be made flush left, or outdented one level from the surrounding code. When either of these options are in effect, the rest of the code is indented one level to make room for the LABEL outdenting. Lines that begin with either kind of comment are not moved.

- Many different styles of indenting of structures and continued lines are available, including a very elaborate mixed indenting style which makes complex SQL commands more readable.
- Additional control of indenting and outdenting with --INDENT and --OUTDENT comments. Useful for making GOTO loops clearer.
- Lines starting with 3 dashes will be forced flush-left. This is an extension of the R:BASE -- comment.
- Smart line wrapping option. Finds words broken across lines and automatically fixes them if the wrapping option is turned on. Dual column wrapping where code wraps wide when not followed by dash comment and wraps short when followed by dash comment. Dash comments following code move to designated column. Wrap columns are user-defined. Lines not to be wrapped can be flagged by appending a semicolon before the dash comment.
- Optional unwrapping
- Optional keyword casing: upper case, lower case
- Optional keyword expansion
- Optional Non-keywords casing: upper case, lower case
- Optional custom mixed casing for non-keywords
- Optional custom mixed casing for variables
- Optional custom word expansions
- Can substitute <= for LE, <> for NE, etc
- Specify files to process with wildcards
- Exclude files to process with wildcards
- Quote: single or double. SET QUOTES tracked to process files correctly
- BLANK, DELIMIT and SEMI can be defined and tracked. These settings are recognized and used by all features.
- Support for SET SEMI ON
- Option to split a SET VAR statement with multiple parts into many separate commands which are easier to debug with TRACE

## 4.2 Why use R:Style?

Plain and simple, it will save you hours of debugging time. There are many mistakes which you can make when writing R:BASE code which are very time consuming to find. Some of these mistakes cause problems only intermittently. Some only cause problems when your code is code-locked. Sometimes the editors we use introduce subtle errors which do not show on the screen but can confuse R:BASE. The list is endless.

Developers who have started using R:Style tell us they would never want to be without R:Style again. We venture to say that R:Style saves at least a couple of hours a week in debugging time.

First of all, R:Style checks the structure of your code to make sure you have balanced quotes, parentheses, IF/ENDIF, WHILE/ENDW, etc. These are all problems which should be fixed before the code is run or traced. It is a lot faster to R:Style a new or modified piece of code, than it is to run the code. It literally takes a few seconds, R:Style is blazing fast. Structural inconsistencies are the kind of problems that cause TRACE to act weirdly or bomb completely. R:Style reports code errors with their line numbers so you can find and fix them quickly. The error report goes to the screen, but the output can be redirected to a file or printer. Only the part of the screen display which you need for debugging will be redirected. This is especially helpful the first time you run your code through R:Style, as there is likely to be a large list the first time.

Secondly, TABS and other hidden characters embedded in the code (except in quoted strings which R:Style protects) can cause erratic behavior of your code in sometimes unpredictable ways. They are difficult for a programmer to find but easy for R:Style to scrub out. Because R:Style can cleanly reformat code to a style pleasing to you, the scrubbing actually will enhance the way your code looks. You have a raft of styling options to achieve this goal.

R:Style finds problems that we are tripping over all the time. It will actually teach you to write cleaner code as you see the errors it flags. In addition, following its advice will make your code more compatible with future versions of R:BASE, which always seem to get less tolerant of non-conforming code.

## 4.3 Configuring R:Style

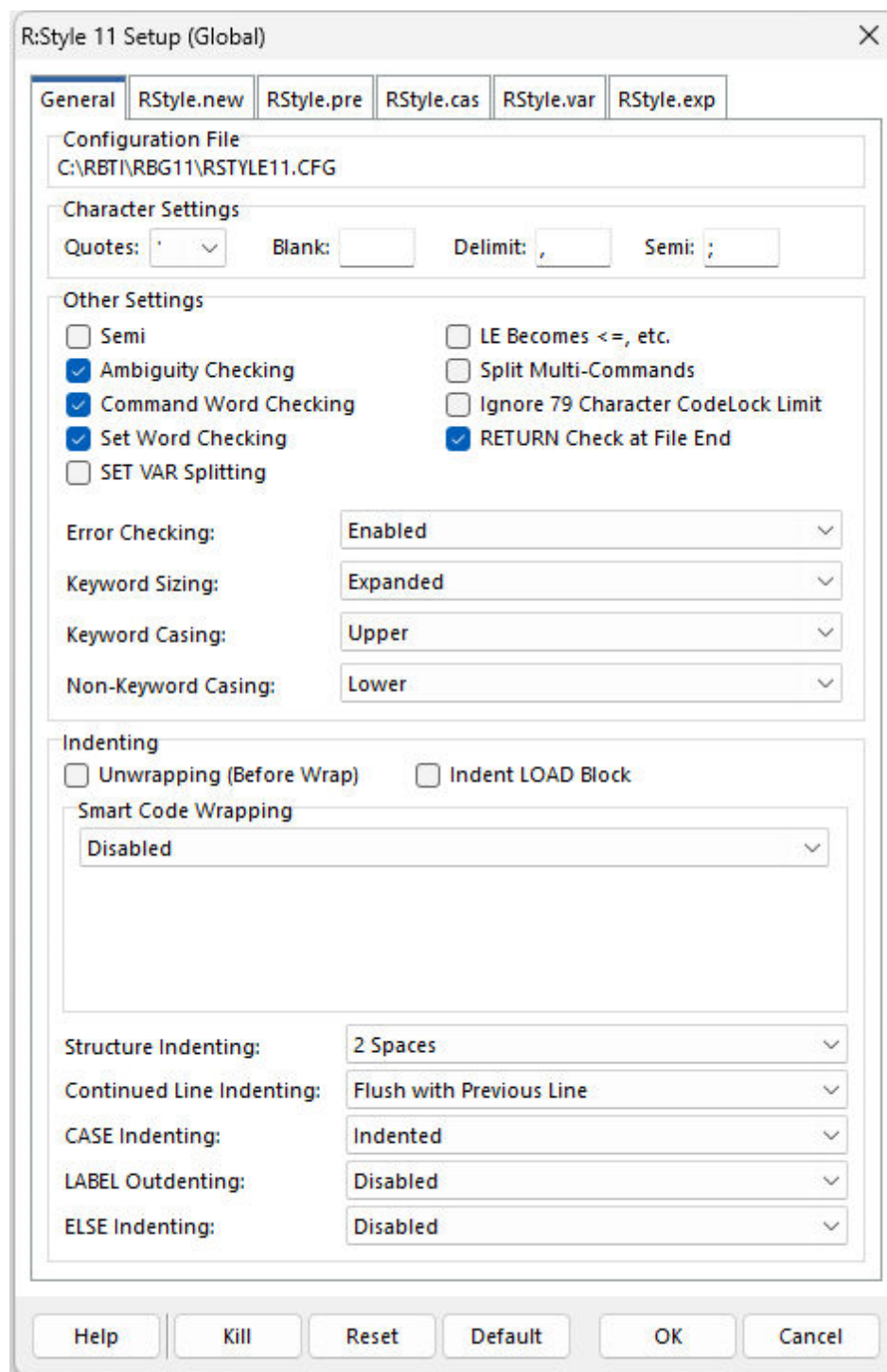
R:Style provides structural error checking, spelling assistance, and structural formatting to help debug your code prior to running it. For more details on what R:Style can do, please see [What does R:Style do?](#)

The R:Style Setup allows you to establish configuration settings based upon the characters used in your command files and how you want R:Style to format your code. Options for the R:Style configuration file(s), RSTYLE11.CFG, consist of Global and Local locations.

- **Global** - forces R:Style to read or create the global configuration file in the default installation directory
- **Local** - forces R:Style to read or create a local configuration file in the working directory
- **Set User-Defined Local R:Style Files Location** - Assigns a user-defined folder for the location of the local R:Style files
- **Reset Local R:Style Files Location** - Resets the local R:Style files folder location to (always be) the current folder

R:Style uses a specific logic to determine what configuration file to use. The default is set to read the local file if it exists. If the local file does not exist then the global file is read. If neither exists a global file is created.

When launching the R:Style Setup, you will be presented with the following configuration screen:



Many R:Style features can be disabled if the results are unwanted for your code.

#### Configuration File

The current R:Style configuration file and path will be listed. The global configuration file in the default installation directory or a local configuration file will be displayed.

#### Character Settings

The QUOTES, BLANK, DELIMIT, and SEMI characters can be configured to match what is used within your command files. R:Style will not let you set the characters to duplicate values. So, if you want to change DELIMIT to ";" then you must change SEMI to something else first.

R:Style checks for the correct QUOTES setting to prevent users from incorrectly altering the code with an improper setting. In addition, R:Style tracks the SET QUOTES command, where if the QUOTES the value is changed, R:Style will expect the value to be restored within the same command file. It is good programming practice that any changes to SET QUOTES affects only code in that same command file.

**Other Settings**

- Semi - When enabled/checked to match when the SEMI operating condition is ON, all lines are considered one command until a semicolon (or the current semi setting) is encountered. No plus characters are used for continuation. SEMI OFF is the standard R:BASE setting.
- Ambiguity Checking - checks for commands that can have multiple meanings such as AUT, AUTO, CAS, CHD, COM, COMM, CON, DEF, DIS, FIL, NEW, NEX, NEXT, PRE, PREV, PRO, REC, RECO
- Command Word Checking - checks for valid commands. If this setting is not checked, then no command checks will be performed and invalid commands such as SETVAR (rather than SET VAR) will not be marked as an error line.
- Set Word Checking - Checks for ambiguous SET commands e.g. "SET AUT ON" will trigger an error because "AUT" can have multiple meanings. The words checked after SET are: AUT, AUTO, AUTOC, AUTOCO, AUTOR, AUTOS, AUTOU, AUTOUP, COL, COM, CUR, EXP, FAS, FAST, LIN, LINE, MAN
- SET VAR Splitting - splits a SET VAR statement with multiple variable definitions into separate SET VAR commands. The separate SET VAR commands can be easier to debug with the Trace Debugger.
- LE Becomes <=, etc. - substitutes letter conditions with their character counterpart; <= for LE, <> for NE, >= for GE, etc.
- Split Multi-Commands - breaks up multi-command lines. If this option is used, indenting will not be preformed when the lines are broken. A second pass through R:Style will correct the indenting.
- Ignore 79 Character CodeLock Limit - specifies to ignore dash comments extending beyond 79 characters, which breaks CodeLock procedures
- RETURN Check at File End - specifies to "not" check for a RETURN at the end of the command file

**Error Checking**

Disabled - disables error checking

Enabled - performs error checking (default)

Read-Only - performs error checking, but does not alter the command file

**Keyword Sizing**

Disabled - disables keyword sizing

Abbreviated - maintains abbreviated keywords

Expanded - expands abbreviated keywords to the full word (default)

R:Style uses an abbreviation list to figure out how to expand keywords to their full length. All abbreviations of command keywords are unique, all abbreviations of SET keywords are unique, and all other keyword abbreviations are unique. Because of this, some of the abbreviations are not the very smallest allowable. In a couple of cases the expansion is not maximum either. One example is VARIABLES. There are some places VARIABLES is illegal but VARIABLE always works, so that is what R:Style uses. In addition, V expands only to VAR and VARI expands to VARIABLE. Because of these constraints, you can process your code all you want without breaking your applications.

**Keyword Casing**

Disabled - disables keyword casing

Lower - applies lower casing to keywords

Upper - applies upper casing to keywords (default)

If R:Style cases any words in your code as if they were keywords, this is an indication that you may run into trouble with that word usage for the application, when you upgrade R:BASE versions. The keyword may not have been a reserved word in a prior release, but is now. For example, the reserved word "over" was added in R:BASE 10.5. If a table or column in a database is named "over" the automatic casing to OVER will take place. It is wise to correct the name in the databases and within command files to avoid using such keywords.

**Non-Keyword Casing**

Disabled - disables non-keyword casing

Lower - applies lower casing to non-keywords, such as variables, table names, component IDs, etc. (default)

Upper - applies upper casing to non-keywords

**Indenting** - R:Style supports many combinations of indenting style. Structures can be indented, and continued lines can carry a different indenting style. Experiment with the configuration to see the various styles available. There is a great deal of flexibility built in to allow you to find a style that you like. Special case indenting can also be accomplished using --INDENT and --OUTDENT comment pairs within command file. These are treated just like WHILE and ENDWHILE pairs. This is useful when you have replaced a WHILE loop with GOTO's and LABEL's and you want the structure to retain it's indenting.

- Unwrapping (Before Wrap) - unwraps continued lines before wrapping them, allowing review of R:Style wrapping
- Indent LOAD Block - indents the lines of data to be loaded into a table
- Smart Code Wrapping - wraps command lines within single or double columns by specifying column widths, left as is with the option disabled. The double column code wrapping is for when where code is on the left and dash comments are on the right. You can even flag lines not to be wrapped by ending the code with a semicolon, before any comments.
- Structure Indenting - provides command structure indenting based upon the number of spaces (default is 2 spaces), code to be flush left, or structure may be left as is with the option disabled.
- Continued Line Indenting - spacing options are available with or without mixed double indenting, and exceptions after IF, WHILE, and ELSE structures. The Mixed Double Indenting works well on complex SQL statements to make them more readable.
- CASE Indenting - CASE structures may be indented or flush with the SWITCH command
- LABEL Outdenting - LABEL commands can be outdented, flushed left, instances can also be left as is with the option disabled
- ELSE Indenting - ELSE structures can be indented, or left as is with the option disabled

*Help* - launches the R:Style chapter of the help

*Kill* - disables all R:Style settings

*Reset* - resets the R:Style setup settings to when the dialog was opened

*Default* - resets the R:Style setup settings to the default values

*OK* - saves any setting changes made

*Cancel* - cancels any setting changes

### **Saving the Configuration**

Once you have adjusted the configuration to your preferences, selecting the OK button will save the configuration when you are done. Pressing OK will save the settings to the RSTYLE11.CFG in the default installation directory. Please refer to this file as the Global configuration since R:Style will always be able to find it if a Local configuration has not been created. If a configuration file already exists, you will be asked if you wish to save the configuration. The path and file name will be displayed for your verification.

### **Testing R:Style**

To test R:Style you can simply save any of your command files before running R:Style, then after the changes and formatting are made, you can close the file without saving the R:Style results. Also, the "Error Checking" option has a Read-Only option available. Use this if you never want your files restyled. This will also disable the hidden character scrubbing that R:Style does.

### **Preventing Alteration of Certain Words**

In some cases, it is not just practical, but highly desirable, to change names used in your database so they do not match any possible keyword or reserved word. In this case you want to prevent R:Style from modifying those words or complaining about their existence. Place these words, one to a line, in the R:Style Setup under the "RStyle.PRE" tab. These words will be cased as they are entered, but will not be altered in any other way. A file, "RSTYLE.PRE", will be created in the directory. A RSTYLE.PRE file will be created in both the Global and Local directories for each R:Style configuration.

### **Customizing Mixed Case Table or Column Names**

To custom case certain words, such as table or column names, place these words, one to a line, in the R:Style Setup under the "RStyle.CAS" tab. These words will be cased as they are entered, but will not be altered in any other way. A file, "RSTYLE.CAS", will be created in the directory. A RSTYLE.CAS file will be created in both the Global and Local directories for each R:Style configuration.

### **Customizing Mixed Case Variable Name**

To custom case variable names, place these words, one to a line, in the R:Style Setup under the "RStyle.VAR" tab. These words will be cased as they are entered, but will not be altered in any other way. A file, "RSTYLE.VAR", will be created in the directory. A RSTYLE.VAR file will be created in both the

Global and Local directories for each R:Style configuration. Since only the first word on each line of RSTYLE.VAR is used, you may also use this file to document your variables.

#### **Simplifying Maintenance**

If you do not want to maintain two files, simply keep all new words in either the RStyle.CAS or RStyle.VAR tabs.

## **4.4 A Bit of Advice**

Always backup your files before processing them with any utility of this sort. No matter how good this utility is, there is always the chance that a subtle bug or hardware failure can destroy your work. Your time is too valuable to waste by ignoring this advice.

## **4.5 Running R:Style**

Always change to your application working directory before running R:Style. While R:Style will handle full paths, it is better to work in one directory at a time. That way you can create a local configuration for that particular directory.

Simply start R:BASE Editor and then click on R:Style Setup icon to create a local configuration for that particular directory. Then, from the Menu Bar, choose **Run: R:Style 11**, or click the R:Style button on "Run" tool bar.

## **4.6 Order of Processing**

As your file is read, each word is checked against the following lists. The first match found determines what will be done with the word.

01. External preprocess list (RSTYLE.PRE)
02. Internal command keyword list if first word in command or next word after DEBUG
03. Internal set keyword list if next word after SET
04. Internal keyword list
05. Internal casing list
06. Internal variable list

If not found yet it is considered a non-keyword.

07. External expansion list (RSTYLE.EXP)
08. External casing list (RSTYLE.CAS)
09. External variable list (RSTYLE.VAR)

If not found, each word is cased as a non-keyword but added to RSTYLE.NEW without alteration. If you check RSTYLE.NEW after each run you will be able to put those words in RSTYLE.CAS or RSTYLE.VAR and case them the way you want if not that way already. If you leave words in RSTYLE.NEW things will slow down as the file grows so check on it occasionally.

Also, words in RSTYLE.NEW will not be used in casing. If you see the case changing to all lower on new words, just move them to the appropriate files and they will be properly cased from then on.

## **4.7 Troubleshooting**

Here are some quick hints to help you around some common pitfalls.

- **Option added to check first word of commands.**

R:BASE allows the following 2 syntax variations:

Syntax: varname = whatever

Syntax: SET VAR varname = whatever



With first word check option on, the first syntax will throw an error while the second will not. The second is more standard syntax. The first would be too complicated to be practical in R:Style. This is why this is optional, so you are not forced to our preferences, but at the same time you lose a very valuable checking feature.

- **Indenting not working the same**

Check your configuration. You may have to fix the indenting and continued line indenting settings.

- **R:Style is misbehaving**

Occasionally, the RSTYLE.CFG will get corrupted and cause R:Style to blow its cool. Delete the offending CFG and recreate it. R:Style should become well behaved once more.

- **Unloaded Structure**

Since R:BASE unloads some of its structure with lines broken in the middle of words, ALWAYS turn on R:Style's unwrapping option the first time you R:Style any code created by R:BASE's UNLOAD command.

- **Lists**

When using the syntax "WHERE colname IN (list)" make sure to quote each value that could be mistaken for a command. V will be expanded to VAR while 'V' will be untouched. Technically, all string values should be quoted even though R:BASE will usually not complain if they are not. The best style is to quote all the elements of the list. Quoted text elements will never be added to RSTYLE.NEW.

- **RSTYLE.NEW**

When R:Style does not know a word it adds it to RSTYLE.NEW. If this file grows very large, R:Style will slow down. Move the words to RSTYLE.CAS or RSTYLE.VAR a priority. Any words that do not belong there are indications of a syntax error on your part (see LISTS above) or a bug in R:Style. You should never get dotted or ampersand variables in this file. If you do, please report the bug so it can be addressed. Please include the offending lines of code. You should get complete filespecs in this file. R:Style should not ever break up a filename or filespec if it is legal syntax.

- **Line Numbers**

R:Style reports errors on the lines in the target file. If processing on a file is aborted for some reason, R:Style restores the original file so the reported line numbers may not match. In this case, fix whatever aborted the processing and try again to find the other errors.

- **Double Line Wrapping**

The short wrapped code will be wrapped at or before the column specified. The following comments will be placed two columns beyond the designated wrapping column.

- **Quotes**

In some cases, when you have 2.11 or earlier code where the single quote is used around column names, R:Style will refuse to process the file. This is because there are more single quotes than double quotes in the file. Simply add enough SET QUOTES """" (that's 4 double quotes) to the beginning of the file so there are more double quotes in the file than single quotes.

- **Convex (Convert Express)**

When using the CONVEX, converter utility by R:BASE Technologies, to convert from 2.11 to 4.5++, make sure to run the resulting code through R:Style with LINE UNWRAPPING turned ON so R:Style will be able to fix all the lines wrapped in the middle of words. Once that is done R:Style will work smoother with the resulting code

Make sure to make a backup of your database, first.

1. Now run the following:

SET NULL -0-

OUTPUT forms.uld  
UNLOAD DATA FROM SYS\_FORMS  
OUT SCR

2. Then run R:Style on the forms.uld
3. Finally, remove the old forms and load the R:Style corrected forms.

DELETE ROWS FROM SYS\_FORMS  
INPUT forms.uld

**Part**

**V**

## 5 Technical Support

Please read over the help documentation at least once before seeking support. We have worked very hard to make the help documentation clear and useful, but concise. It is suggested that you reread these instructions once you have become accustomed to using the software, as new uses will become apparent.

If you have further questions, and cannot find the answers in the documentation, you can obtain information from the below sources:

- Email our Technical Support Staff at: [support@rbase.com](mailto:support@rbase.com)
- Access the R:BASE Technologies Support home page online at <https://www.rbase.com/support>

You may be required to purchase a technical support plan. Several support plans are available to suit the needs of all users. [Available Technical Support Plans](#)

Please be prepared to provide the following:

- The product registration number, which is located on the invoice/order slip for the purchased product
- The type of operating system and hardware in use
- Details regarding your operating environment; such as available memory, disk space, your version of R:BASE, local area network, special drivers, related database structures, application files, and other files that are used or accessed by your application

All provide information will be used to better assist you.

R:BASE Technologies has a number of different services available for R:BASE products. As a registered user, you will receive information about new features for R:BASE and other R:BASE Technologies products. Please remember to register your software. <https://www.rbase.com/register/>

**Part**

**VI**

## 6 Useful Resources

- . R:BASE Home Page: <https://www.rbase.com>
- . Up-to-Date R:BASE Updates: <https://www.rbaseupdates.com>
- . Current Product Details and Documentation: <https://www.rbase.com/rbg11>
- . Support Home Page: <https://www.rbase.com/support>
- . Product Registration: <https://www.rbase.com/register>
- . Official R:BASE Facebook Page: <https://www.facebook.com/rbase>
- . Sample Applications: <https://www.razzak.com/sampleapplications>
- . Technical Documents (From the Edge): <https://www.razzak.com/fte>
- . Education and Training: <https://www.rbase.com/training>
- . Product News: <https://www.rbase.com/news>
- . Upcoming Events: <https://www.rbase.com/events>
- . R:BASE Online Help Manual: <https://www.rbase.com/support/rsyntax>
- . Form Properties Documentation: <https://www.rbase.com/support/FormProperties.pdf>
- . R:BASE Beginners Tutorial: <https://www.rbase.com/support/rtutorial>
- . R:BASE Solutions (Vertical Market Applications): <https://www.rbase.com/products/rbasesolutions>

**Part**

---

**VII**

## 7 Feedback

### **Suggestions and Enhancement Requests:**

From time to time, everyone comes up with an idea for something they'd like a software product 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 an ideal 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](mailto: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 message displayed
- The operating system 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](mailto:rdcc@rbase.com).



# Index

## - [ -

[Alt] 30  
[Ctrl] 30

## - A -

active cursor 55  
Advice 68  
ASCII 22  
Auto Indent 52

## - B -

Back Unindents 52  
background color 55  
background style 55  
Beyond File End 52  
block 55  
Blocks 52  
bookmark 20  
Bookmarks 51  
Break Word 51  
Break Word at Right Margin 51

## - C -

calculator 21  
Caret 51  
Case 64  
casing 64  
character 22  
check 45  
Clipboard History 23  
close 29  
code 31  
code fragment 31  
Code Template 31  
Code Templates 20  
command 32  
Command Builder 32  
COMMAND.INI 32

comment 32  
Comments 38  
Common 38  
Compare 24  
Compare by Content 57  
Context 24, 57  
Current Directory 25  
Cursor Always on Tabs 52  
Cursor on Tabs 52  
Cursor Style 51

## - D -

Data Dictionary 21, 34  
decryption 47  
Disable Drag 51  
Display 53  
Double Click Line 51  
Drag 51  
Draw Gutter 51  
Draw Line 51  
Draw Line Bookmarks 51  
Draw Right Margin 51

## - E -

edit 38  
Edit Syntax Schemes 20  
Editor 17  
Editor Settings 20  
encryption 47  
EOL 52  
Error 26  
ErrorMessages 26  
Exclude Read-Only 52  
expand 64  
expansion 64  
extensions 38, 44

## - F -

feedback 76  
file 29  
file association 44  
File Picker 26  
File Structure 27  
Fill 52

Find 21, 36, 37, 48, 52  
find files 35  
Find in Files 21, 35  
Find Text 52  
Find Text at Cursor 52  
Folder Favorites 25  
Force Cut/Copy 52  
Full Line 38  
function 32

## - G -

General 51  
Group Undo 52  
GUID 21  
Gutter 51, 53  
Gutter Color 53  
Gutter Width 53

## - H -

hot keys 30  
HTML 19

## - I -

Identifiers 38  
Indent 52, 64  
insert 21  
install 14

## - K -

key 64  
Key Assignments 20, 38  
key combinations 30  
keyword 64  
Keywords 38

## - L -

license 11  
Limit EOL 52  
Limit Line Numbers 51  
Line 64  
Line Highlight 55

Line Number Color 53  
Line Numbers 51  
Line Separator 53

## - M -

maintenance 64  
map 22  
Margin 51  
Menu 17  
Messages 26  
multi-view 29

## - N -

next 29  
nonkey 64  
non-key 64  
non-keyword 64  
Notepad 51  
Notepad Cursor Style 51  
Number Color 53  
Numbers 38

## - O -

Only Text 51  
Optional Fill 52  
order 68  
Outdent 64  
Overwrite 51  
Overwrite Blocks 52  
Overwrite Caret 51

## - P -

password 47  
Persistent Blocks 52  
previous 29  
print 19  
printer font 57  
programming 31

## - R -

R:Style 21, 61, 62, 63, 64, 68

R:Style Setup 20  
read only 20  
Read-Only 52  
redo 20  
regular expression 36, 37, 48  
replace 21, 36, 37, 48  
Right Margin 51, 53  
RStyle 61, 62, 63, 68  
RTF 19  
Run 19, 68

## - S -

Save As 47  
scheme 38  
search 21, 35  
Select Found Text 52  
Select Only Text 51  
selected text 55  
Separato 53  
setup 64  
shell integration 20, 44  
shortcut keys 38  
Show Error Message Panel 53  
simultaneous 29  
Smart 52  
Smart Tab 52  
spell 45  
spell check 21  
Spell Checker Setup 20  
spellcheck 45  
split 64  
String 52  
Strings 38  
Structure 27  
Structure Panel 53  
Structure Toolbar 58  
style 38  
support 15  
synchronize 29  
sync-scroll 29  
syntax 38  
system requirements 14

## - T -

tab 29

Tab Stops 53  
template 31, 32  
Text 24, 36, 37, 48, 57  
Text Match 55  
thesaurus 21, 46  
Thumb 51  
Thumb Tracking 51  
Tracking 51  
trouble 68

## - U -

Undo 20, 52  
Unindents 52  
Unwrap 64  
Use Tab Character 52  
User-Defined Mask 26

## - W -

Window 17  
Word Wrap 51  
Wrap 51, 64

Back Cover